



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI MOBILE
ANDROID SISTEM KEHADIRAN MAHASISWA
MELALUI PENCOCOKAN WAJAH DENGAN
MENGUNAKAN LIBRARY ANDROID FACE
RECOGNITION WITH DEEP LEARNING STUDI
KASUS JURUSAN TEKNIK INFORMATIKA ITS**

RONALD GUNAWAN RADITYATAMA
NRP. 5113 100 112

Dosen Pembimbing 1
Sarwosri, S.Kom., M.T.

Dosen Pembimbing 2
Dwi Sunaryono, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI MOBILE
ANDROID SISTEM KEHADIRAN MAHASISWA
MELALUI PENCOCOKAN WAJAH DENGAN
MENGUNAKAN LIBRARY ANDROID FACE
RECOGNITION WITH DEEP LEARNING STUDI
KASUS JURUSAN TEKNIK INFORMATIKA ITS**

**RONALD GUNAWAN RADITYATAMA
NRP. 5113 100 112**

**Dosen Pembimbing 1
Sarwosri, S.Kom., M.T.**

**Dosen Pembimbing 2
Dwi Sunaryono, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - KI141502

**DEVELOPMENT OF STUDENT ATTENDANCE
SYSTEM THROUGH FACE VERIFICATION USING
ANDROID FACE RECOGNITION WITH DEEP
LEARNING LIBRARY IN ANDROID MOBILE-
BASED APPLICATION FOR DEPARTMENT OF
INFORMATICS ITS CASE STUDY**

RONALD GUNAWAN RADITYATAMA
NRP. 5113 100 112

Dosen Pembimbing 1
Sarwosri, S.Kom., M.T.

Dosen Pembimbing 2
Dwi Sunaryono, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI MOBILE ANDROID SISTEM KEHADIRAN MAHASISWA MELALUI PENCOCOKAN WAJAH DENGAN MENGGUNAKAN LIBRARY ANDROID FACE RECOGNITION WITH DEEP LEARNING STUDI KASUS JURUSAN TEKNIK INFORMATIKA ITS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

RONALD GUNAWAN RADITYATAMA

NRP. 5113 100 112

Disetujui oleh Pembimbing Tugas Akhir

1. Sarwosri, S.Kom., M.T.
(NIP. 197608092001122001)
2. Dwi Sunaryono, S.Kom., M.Kom.
(NIP. 197205281997021001)



**SURABAYA
JULI, 2017**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI MOBILE ANDROID SISTEM KEHADIRAN MAHASISWA MELALUI PENCOCOKAN WAJAH DENGAN MENGGUNAKAN LIBRARY ANDROID FACE RECOGNITION WITH DEEP LEARNING STUDI KASUS JURUSAN TEKNIK INFORMATIKA ITS

Nama Mahasiswa : Ronald Gunawan Radityatama
NRP : 5113 100 112
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Sarwosri, S.Kom., MT.
Dosen Pembimbing 2 : Dwi Sunaryono, S.Kom., M.Kom.

ABSTRAK

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Sistem kehadiran ini dapat menjadi bukti bahwa seorang mahasiswa telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Di Jurusan Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran.

Dalam tugas akhir ini dirancang sebuah sistem kehadiran mahasiswa yang mengimplementasikan metode pencocokan atau pengenalan wajah (face recognition) ke dalam aplikasi mobile. Aplikasi ini diimplementasikan dengan menggunakan pustaka (library) Android Face Recognition with Deep Learning dimana didalamnya terdapat algoritma dan pustaka-pustaka lainnya yang berhubungan dengan proses pengenalan wajah. Implementasi dari metode ini diharapkan dapat mengatasi kemungkinan terjadinya kecurangan dalam pencatatan kehadiran.

Hasil pengujian dari Implementasi metode pengenalan wajah ke dalam aplikasi mobile menunjukkan hasil yang cukup

memuaskan. Hal ini terlihat dari hasil pengujian kegunaan dari aplikasi ini yang memperoleh hasil diatas 4 dari skala 1 sampai 6 untuk masing-masing kegunaan yang diuji. Dari penilaian ini dapat disimpulkan bahwa aplikasi ini cukup fleksibel, layak dan mudah untuk digunakan.

Kata kunci: android, face recognition, sistem kehadiran mahasiswa

DEVELOPMENT OF STUDENT ATTENDANCE SYSTEM THROUGH FACE VERIFICATION USING ANDROID FACE RECOGNITION WITH DEEP LEARNING LIBRARY IN ANDROID MOBILE- BASED APPLICATION FOR DEPARTMENT OF INFORMATICS ITS CASE STUDY

Student Name : Ronald Gunawan Radityatama
NRP : 5113 100 112
Major : Informatics Department FTIf - ITS
Supervisor I : Sarwosri, S.Kom., MT.
Supervisor II : Dwi Sunaryono, S.Kom., M.Kom.

ABSTRACT

Student attendance system is the process of recording student attendance in class. This attendance system can be proven that a student has been in the classroom. The process of recording student attendance can be done in various ways. In the Department of Informatics ITS, the process of recording student attendance is done manually. Students sign the student attendance sheets that have been prepared beforehand. This method leads to weaknesses. The weakness is the possibility of fraud in attendance recording process.

This final project is created to design a student attendance system that implements face recognition method into mobile application. This application is implemented by using Android Face Recognition with Deep Learning library. This library consists of several algorithms and other libraries related to face recognition process. Implementation of face recognition is expected to address the possibility of fraud in attendance.

The test results of the implementation of face recognition method into mobile applications show satisfactory results. The value of the results of usability testing app to get results above 4

on the scale of 1 to 6 for each usability test. From the test results it can be concluded that the application is sufficiently flexible, feasible and easy to use.

Key words: android, face recognition, student attendance system

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah yang Maha Esa Karena atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“RANCANGBANGUN APLIKASI MOBILE ANDROID SISTEM KEHADIRAN MAHASISWA MELALUI PENCOCOKAN WAJAH DENGAN MENGGUNAKAN LIBRARY ANDROID FACE RECOGNITION WITH DEEP LEARNING STUDI KASUS** kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama proses pengerjaan **JURUSAN TEKNIK INFORMATIKA ITS”**.

Pengerjaan tugas akhir ini penulis lakukan untuk memenuhi salah satu syarat memperoleh nformatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih tugas akgelar Sarjana Komputer di Program Studi S-1 Jurusan Teknik Ihir ini hingga selesai, antara lain:

1. Allah yang Maha Esa atas segala karunia dan rahmat-Nya yang telah diberikan selama ini.
2. Keluarga penulis, Ayah, Ibu, dan saudara kandung serta keluarga yang tidak dapat penulis tuliskan satu per satu yang telah memberi dukungan moral, material, dan juga doa untuk penulis.
3. Ibu Sarwosri, S.Kom., M.T. selaku dosen pembimbing I yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.
4. Bapak Dwi Sunaryono S.Kom., M.Kom. selaku dosen pembimbing II yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.
5. Bapak Dr. Eng. Darlis Herumurti, S.Kom.,M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Dr. Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah banyak memberikan ilmu kepada penulis.

6. Teman-teman Administrator Laboratorium Algoritma Pemrograman, penghuni ruang Himpunan Mahasiswa Teknik Computer-Informatika ITS angkatan 2013 yang telah menjadi sahabat dan keluarga selama penulis menimba ilmu di Teknik Informatika ITS.
7. Teman-teman angkatan 2013 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, serta adik-adik angkatan 2014 dan 2015 yang membuat penulis untuk selalu belajar.
8. Saudari Chynthia Laurencia yang telah memberi dukungan moral, dan juga doa untuk penulis.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam tugas akhir ini. Oleh karena itu, penulis menerima dengan rendah hati kritik dan saran untuk pembelajaran dan perbaikan ke depannya. Semoga tugas akhir ini dapat memberikan manfaat yang sebaik-baiknya.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DATAR TABEL.....	xix
DAFTAR KODE SUMBER.....	xx
BAB I PENDAHULUAN.....	xxii
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Metodologi Pembuatan Tugas Akhir	4
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	6
BAB II TINJAUAN PUSTAKA.....	9
2.1 Penelitian Terkait	9
2.2 Teknik Informatika ITS.....	10
2.3 Sistem Kehadiran	11
2.4 Face Recognition	12
2.5 Penggunaan Pustaka Android Face Recognition with Deep Learning.....	13
2.5.1 Penggunaan OpenCV Library.....	15
2.5.1.1 Penggunaan Algoritma Tahap Deteksi Wajah.	16
2.5.1.2 Penggunaan Algoritma Tahap Preprocessing..	17
2.5.1.3 Penggunaan Algoritma Tahap Ekstraksi Fitur	18
2.5.2 Penggunaan LIBSVM.....	18
2.5.2.1 Penggunaan Algoritma Tahap Klasifikasi	19
2.6 APL.....	20
2.7 JSON.....	20
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	21
3.1 Analisis	21

3.1.1	Analisis Permasalahan	21
3.1.2	Deskripsi Umum Sistem	23
3.1.3	Spesifikasi Kebutuhan Perangkat Lunak	24
3.1.3.1	Kebutuhan Fungsional.....	24
3.1.3.2	Kebutuhan Non-Fungsional	25
3.1.3.3	Aktor	25
3.1.3.4	Kasus Penggunaan.....	25
3.1.3.4.1	Mendaftarkan Wajah (UC-001)	26
3.1.3.4.2	Melakukan Presensi (UC-002).....	31
3.2	Perancangan.....	35
3.2.1	Perancangan Data.....	35
3.2.2	Perancangan Arsitektur Perangkat Lunak.....	36
3.2.3	Perancangan Diagram Kelas.....	37
3.2.4	Perancangan Antarmuka Pengguna	37
3.2.4.1	Rancangan Halaman Antarmuka Menu Utama Verifikasi Wajah	38
3.2.4.2	Rancangan Halaman Antarmuka Kelola Data Wajah.....	39
3.2.4.3	Rancangan Halaman Antarmuka Mendaftarkan Data Wajah.....	41
3.2.4.4	Rancangan Halaman Antarmuka Melatih Data Wajah.....	42
3.2.4.5	Rancangan Halaman Antarmuka Melakukan Presensi	43
3.2.5	Perancangan Proses.....	44
3.2.5.1	Rancangan Alir Proses Utama Perangkat Lunak	44
3.2.5.2	Rancangan Alir Proses Mengelola Data Wajah	46
3.2.5.3	Rancangan Alir Proses Daftarkan Data Wajah.....	47
3.2.5.4	Rancangan Alir Proses Latih Data Wajah	49
3.2.5.5	Rancangan Alir Proses Pencocokan Wajah Dengan Pustaka Android Face Recognition With Deep Learning.....	51
BAB IV IMPLEMENTASI.....		55

4.1	Lingkungan Implementasi Perangkat Lunak	55
4.2	Implementasi Antarmuka Pengguna	56
4.2.1	Implementasi Halaman Antarmuka Menu Utama Verifikasi Wajah	56
4.2.2	Implementasi Halaman Antarmuka Kelola Data Wajah	57
4.2.3	Implementasi Halaman Antarmuka Mendaftarkan Data Wajah	59
4.2.4	Implementasi Halaman Antarmuka Melatih Data Wajah	59
4.2.5	Implementasi Halaman Antarmuka Presensi	62
4.3	Implementasi Kasus Penggunaan dan Alir Proses	64
4.4	Implementasi Kasus Penggunaan Mendaftarkan Data Wajah	64
4.4.1	Implementasi Kasus Penggunaan Melakukan Presensi	69
4.4.2	Implementasi Alir Proses Latih (training) Data Wajah	72
BAB V PENGUJIAN DAN EVALUASI		81
5.1	Lingkungan Pengujian	81
5.2	Pengujian Fungsionalitas	81
5.2.1	Pengujian Fungsionalitas Mendaftarkan Data Wajah	82
5.2.2	Pengujian Fungsionalitas Melakukan Presensi	85
5.3	Pengujian Kegunaan	88
5.3.1	Pengujian Kegunaan Skenario Kelas A	92
5.3.2	Pengujian Kegunaan Skenario Kelas B	92
5.3.3	Pengujian Kegunaan Skenario Kelas C	93
5.4	Pengujian Waktu Eksekusi	95
5.5	Pengujian Kapasitas Pengolahan Data Wajah	97
5.6	Evaluasi Pengujian Fungsionalitas	99
5.7	Evaluasi Pengujian Kegunaan	100
5.8	Evaluasi Pengujian Waktu Eksekusi	100
5.9	Evaluasi Pengujian Kapasitas Pengolahan Data Wajah	101

BAB VI KESIMPULAN DAN SARAN.....	103
6.1 Kesimpulan.....	103
6.2 Saran.....	103
DAFTAR PUSTAKA.....	105
LAMPIRAN	107
BIODATA PENULIS.....	111

DAFTAR GAMBAR

Gambar 2.1	Arsitektur Aplikasi Pustaka Android Face Recognition with Deep Learning (1).....	13
Gambar 2.2	Arsitektur Aplikasi Pustaka Android Face Recognition with Deep Learning (2).....	14
Gambar 2.3	Arsitektur Aplikasi Pustaka Android Face Recognition with Deep Learning (3).....	14
Gambar 2.4	Hyperplane yang optimal memisahkan dua kelas (+) dan (-) dalam SVM.....	19
Gambar 3.1	Deskripsi Umum Sistem.....	23
Gambar 3.2	Diagram Kasus Penggunaan.....	26
Gambar 3.3	Diagram Aktivitas UC-001.....	28
Gambar 3.4	Diagram Sekuens UC-001.....	29
Gambar 3.5	Diagram Kolaborasi Objek UC-001	30
Gambar 3.6	Diagram Aktivitas UC-002.....	32
Gambar 3.7	Diagram Sekuens UC-002.....	33
Gambar 3.8	Diagram Kolaborasi Objek UC-002	34
Gambar 3.9	Contoh Gambar Wajah Terambil Dan Disimpan	36
Gambar 3.10	Diagram Arsitektur Sistem	37
Gambar 3.11	Rancangan Halaman Antarmuka Menu Utama Verifikasi Wajah	39
Gambar 3.12	Rancangan Halaman Antarmuka Kelola Data Wajah	39
Gambar 3.13	Rancangan Halaman Antarmuka Tambah Data Wajah.....	42
Gambar 3.14	Rancangan Halaman Antarmuka Melakukan Presensi	43
Gambar 3.15	Rancangan Halaman Antarmuka Training Data Set Wajah.....	44
Gambar 3.16	Diagram Alir Proses Utama Perangkat Lunak	45
Gambar 3.17	Diagram Alir Mengelola Data Wajah	47
Gambar 3.18	Diagram Alir Daftarkan Data Wajah	48
Gambar 3.19	Diagram Alir Latih Data Wajah	50
Gambar 3.20	Diagram Alir Proses Pencocokan Wajah.....	52

Gambar 3.21 Diagram Kelas	53
Gambar 4.1 Implementasi Halaman Antarmuka Menu Utama Verifikasi Wajah (sebelum data wajah dilatih)	56
Gambar 4.2 Implementasi Halaman Antarmuka Menu Utama Verifikasi Wajah (setelah data wajah dilatih)	56
Gambar 4.3 Implementasi Halaman Antarmuka Kelola Data Wajah (1)	58
Gambar 4.4 Implementasi Halaman Antarmuka Kelola Data Wajah (2).....	58
Gambar 4.5 Implementasi Halaman Antarmuka Kelola Data Set (3)	60
Gambar 4.6 Implementasi Halaman Antarmuka Kelola Data Set (4)	60
Gambar 4.7 Implementasi Halaman Antarmuka Tambah Data Set Wajah (Tidak Terdeteksi)	60
Gambar 4.9 Implementasi Halaman Antarmuka Tambah Data Set Wajah (Terdeteksi)	61
Gambar 4.10 Implementasi Halaman Antarmuka Training Data Set Wajah	61
Gambar 4.11 Implementasi Halaman Antarmuka Presensi (Wajah Tidak Ditemukan)	62
Gambar 4.12 Implementasi Halaman Antarmuka Presensi (Wajah Terdeteksi dan Cocok)	63
Gambar 4.13 Implementasi Halaman Antarmuka Presensi (Konfirmasi Kehadiran).....	63
Gambar 5.1 Tampilan Kelola Data Wajah Keadaan Awal.....	84
Gambar 5.2 Memilih Pilihan Daftarkan Data Wajah.....	84
Gambar 5.3 Memposisikan Wajah Hingga Terdeteksi oleh Aplikasi	84
Gambar 5.4 Tampilan Verifikasi Wajah	86
Gambar 5.5 Menekan Pilihan Verifikasi Kehadiran.....	86
Gambar 5.6 Tampilan Presensi dan Mendeteksi Wajah	87
Gambar 5.7 Tampilan Dialog Konfirmasi Presensi.....	87
Gambar 5.8 Tampilan Status Presensi	88
Gambar 5.9 Data Wajah Dalam 1 Kelas	98

DAFTAR TABEL

Tabel 2.1 Hasil Pengujian Penggunaan Metode Ekstraksi Fitur dan Klasifikasi dari pustaka Android Face Recognition with Deep Learning	15
Tabel 3.1 Kebutuhan Fungsional	24
Tabel 3.2 Kasus Penggunaan	25
Tabel 3.3 Spesifikasi Kasus Penggunaan UC-001.....	26
Tabel 3.4 Spesifikasi Kasus Penggunaan UC-002.....	31
Tabel 3.5 Atribut Antarmuka Menu Utama Verifikasi Wajah	38
Tabel 3.6 Atribut Antarmuka Kelola Data Wajah	40
Tabel 3.7 Atribut Antarmuka Tambah Data Wajah.....	41
Tabel 3.8 Atribut Antarmuka Latih Data Wajah	42
Tabel 3.9 Atribut Antarmuka Melakukan Presensi.....	43
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	55
Tabel 4.2 Penjelasan Kode Sumber 4.1.....	66
Tabel 4.3 Penjelasan Kode Sumber 4.2.....	69
Tabel 4.4 Penjelasan Kode Sumber 4.3.....	71
Tabel 4.5 Penjelasan Kode Sumber 4.4.....	73
Tabel 4.6 Penjelasan Kode Sumber 4.5.....	76
Tabel 4.7 Penjelasan Kode Sumber 4.6.....	78
Tabel 4.8 Penjelasan Kode Sumber 4.7.....	80
Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak	81
Tabel 5.2 Pengujian Fungsionalitas Mendaftarkan Data Wajah .	82
Tabel 5.3 Pengujian Fungsionalitas Melakukan Presensi.....	85
Tabel 5.5 Kuesioner Pengujian Kegunaan.....	89
Tabel 5.6 Daftar Penguji Aplikasi	90
Tabel 5.7 Hasil Penilaian Kuesioner Kelas A.....	92
Tabel 5.8 Hasil Penilaian Kuesioner Kelas B	93
Tabel 5.9 Hasil Penilaian Kuesioner Kelas C	93
Tabel 5.10 Hasil Penilaian Rata-Rata Kuesioner Seluruh Kelas .	94
Tabel 5.11 Hasil Pengujian Waktu Eksekusi	96
Tabel 5.12 Rincian Kapasitas Data Wajah Tersimpan.....	98

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Mengambil wajah yang terdeteksi oleh kamera	66
Kode Sumber 4.2 Mengunggah data wajah ke sistem server.....	69
Kode Sumber 4.3 Memproses pengenalan wajah.....	71
Kode Sumber 4.4 Melatih data wajah mahasiswa dengan data wajah pembandingan di penyimpanan internal smartphone (1).....	73
Kode Sumber 4.5 Melatih data wajah mahasiswa dengan data wajah pembandingan di penyimpanan internal smartphone (2).....	75
Kode Sumber 4.6 Melatih data wajah mahasiswa dengan data wajah pembandingan di penyimpanan internal smartphone (3).....	78
Kode Sumber 4.7 Melatih data wajah mahasiswa dengan data wajah pembandingan di penyimpanan internal smartphone (4).....	80

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Bab ini menjelaskan garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan, batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Sistem kehadiran ini dapat menjadi bukti bahwa seorang mahasiswa telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Di Jurusan Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran. Kecurangan ini sering disebut titip absen. Titip absen merupakan kejadian di mana mahasiswa tercatat hadir pada kertas daftar kehadiran kelas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Dengan adanya celah kelemahan ini, mahasiswa lain dapat melakukan tanda tangan untuk mahasiswa lain. Oleh karena itu, diperlukan suatu sistem yang dapat digunakan untuk mengatasi kecurangan dalam pencatatan kehadiran mahasiswa di dalam kelas.

Perkembangan teknologi informasi di dunia semakin cepat, khususnya pada teknologi perangkat bergerak (*mobile*). Akses dan pengambilan informasi juga semakin mudah dan cepat diakses melalui penggunaan perangkat bergerak maupun web. Perkembangan ini juga mempengaruhi cara seseorang dalam melakukan proses verifikasi untuk masuk ke dalam sebuah sistem. Verifikasi merupakan sebuah proses yang wajib ada dimana diperlukan sebuah kebenaran data/informasi misalnya verifikasi keaslian identitas. Proses verifikasi ini sudah banyak diterapkan seiring dengan perkembangan teknologi informasi oleh perusahaan, birokrasi, dan lain-lain. Proses verifikasi identitas yang

sering digunakan adalah menggunakan alat deteksi sidik jari (*fingerprint*). Namun kelemahan dari penerapan ini adalah kurangnya fleksibilitas dari proses verifikasi ini. Fleksibilitas yang dimaksud adalah teknologi yang digunakan tidak bisa berpindah tempat dan diakses kapanpun. Oleh karena itu, tugas akhir ini akan mengimplementasikan sebuah aplikasi mobile sistem kehadiran berbasis Android. Implementasi pada aplikasi mobile diharapkan mampu menambah fleksibilitas dari sistem kehadiran yang sudah ada saat ini. Aplikasi kehadiran yang akan dibuat akan dibatasi ruang lingkup penggunaannya. Pengguna dari aplikasi ini adalah mahasiswa Teknik Informatika ITS.

Sistem kehadiran pada aplikasi ini mengimplementasikan metode pencocokan wajah (*face recognition*) ke dalam aplikasi mobile. Aplikasi ini diimplementasikan dengan menggunakan pustaka (*library*) Android Face Recognition with Deep Learning dimana didalamnya terdapat algoritma dan pustaka-pustaka lainnya yang berhubungan dengan proses pengenalan wajah. Secara umum, Prosedur pencocokan wajah dibagi menjadi dua tahapan. Tahap pertama adalah deteksi wajah (*face detection*), yaitu tahap di mana aplikasi mencari wajah dalam gambar tangkapan kamera untuk memastikan adanya wajah seseorang yang tertangkap. Tahap kedua adalah tahap identifikasi atau pengenalan wajah (*face recognition*). Pada tahap ini dilakukan perbandingan hasil ekstraksi fitur wajah yang terdeteksi dengan gambar wajah yang tersimpan dalam *database*. Perbandingan ini dilakukan supaya sistem mendapatkan hasil identifikasi identitas seseorang dari wajah yang terdeteksi. Pemberian identitas ini dilakukan dengan melakukan *training* terhadap aplikasi sebelumnya. Langkah *training* melibatkan pendeteksian wajah yang kemudian hasil deteksi ini diolah dengan menggunakan beberapa teknik pemrosesan gambar supaya dapat diterima dengan baik oleh aplikasi, setelah itu mengajarkan sistem siapa identitas wajah yang terdeteksi dengan menggunakan algoritma *training* yang juga telah disediakan oleh pustaka *face recognition* yang digunakan dalam tugas akhir ini.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara merancang sistem kehadiran mahasiswa melalui pencocokan foto wajah dalam aplikasi mobile Android?
2. Bagaimana cara menerapkan pustaka (*library*) Android Face Recognition with Deep Learning pada perancangan aplikasi mobile untuk pencocokan foto wajah?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Perangkat lunak dapat berjalan pada perangkat mobile dengan spesifikasi versi minimal Android 4.3 (Jellybean / API 18) berkapasitas RAM 2 GB dan resolusi kamera depan 2 MP.
2. Input perangkat lunak berupa pengambilan foto wajah dari *smartphone* milik mahasiswa kemudian foto wajah dikirimkan ke *server* untuk disimpan.
3. Pengolahan foto wajah dilakukan dalam aplikasi *client* atau *smartphone* milik mahasiswa.
4. Data set wajah pembandingan sudah ditentukan dan dimasukkan di dalam APK Android.
5. Proses pengenalan foto wajah dengan pustaka (*library*) Android Face Recognition with Deep Learning dapat berjalan dengan optimal pada kondisi ruang pencahayaan yang cukup.

1.4 Tujuan

Tugas akhir ini mempunyai beberapa tujuan, yaitu sebagai berikut:

1. Merancang sistem kehadiran mahasiswa melalui pencocokan foto wajah dalam aplikasi mobile Android.
2. Menerapkan pustaka (*library*) Android Face Recognition with Deep Learning pada perancangan aplikasi *mobile* untuk pencocokan foto wajah.

1.5 Manfaat

Hasil dari pengerjaan tugas akhir ini memiliki manfaat untuk memperbaharui sistem kehadiran mahasiswa dari sistem manual melalui tanda tangan pada kertas daftar kehadiran mahasiswa menjadi sistem *online* dengan menggunakan metode pencocokan foto wajah pada aplikasi mobile. Dengan demikian, diperolehnya data yang benar tentang kehadiran mahasiswa di kelas akan lebih jelas dan terperinci. Sistem *online* ini juga dapat membantu proses pemantauan dari pihak administrasi kampus, dosen, serta orangtua terhadap mahasiswa atau anaknya.

1.6 Metodologi Pembuatan Tugas Akhir

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang perencanaan “Rancang Bangun Aplikasi Mobile Android Sistem Kehadiran Mahasiswa Melalui Pencocokan Wajah Dengan Menggunakan Library Android Face Recognition with Deep Learning Studi Kasus Jurusan Teknik Informatika ITS”.

Proposal tugas akhir ini terdiri dari deskripsi pendahuluan yang menjabarkan latar belakang dan rumusan masalah yang mendasari dibangunnya aplikasi ini, batasan masalah dalam pembangunan aplikasi ini, serta tujuan dan manfaat yang diharapkan dapat dicapai dengan dibangunnya aplikasi ini. Selain itu, pada proposal tugas akhir ini juga terdapat tinjauan pustaka yang menjelaskan teori-teori yang menjadi dasar pembuatan tugas akhir ini, ringkasan isi tugas akhir yang menggambarkan secara umum aplikasi yang akan dibangun dan algoritma yang digunakan, serta bagian metodologi dari penyusunan proposal tugas akhir ini.

2. Studi literatur

Studi literatur yang dilakukan dalam pengerjaan tugas akhir ini adalah mengenai implementasi penggunaan pustaka (*library*) Android Face Recognition with Deep Learning serta

algoritma-algoritma yang diperlukan untuk perancangan metode pencocokan wajah pada perangkat *mobile* Android. Studi literatur diambil dari buku, internet, maupun materi mata kuliah yang berhubungan dengan metode yang digunakan.

3. Analisis dan desain perangkat lunak

Analisis kebutuhan dan perancangan sistem dilakukan untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Tahap desain meliputi arsitektur perangkat lunak yang digunakan, desain kelas-kelas yang terlibat dalam aplikasi, desain antarmuka, serta diagram-diagram yang mendukung pendeskripsian sistem aplikasi.

4. Implementasi perangkat lunak

Pengembangan aplikasi dalam tugas akhir akan menggunakan bahasa pemrograman Java, XML, dengan kaskas bantu IDE Android Studio. Selain itu, digunakan juga pustaka (*library*) Android Face Recognition with Deep Learning untuk pembangunan metode pencocokan wajah.

5. Pengujian dan Evaluasi

Pengujian dan evaluasi aplikasi perangkat lunak hasil dari tugas akhir ini akan diujicobakan pada mahasiswa Teknik Informatika ITS yang memiliki *smartphone* Android dengan fitur kamera depan. Sebelum pengujian, aplikasi sistem kehadiran mahasiswa akan di-*install* terlebih dahulu di *smartphone* mahasiswa.

6. Penyusunan buku tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan

- a. Latar Belakang
- b. Rumusan Masalah

- c. Batasan Tugas Akhir
- d. Tujuan
- e. Manfaat
- f. Metodologi Pembuatan Tugas Akhir
- g. Sistematika Penulisan Laporan Tugas Akhir
2. Tinjauan Pustaka
3. Analisis dan Perancangan Sistem
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu metodologi yang digunakan dan sistematika penulisan laporan akhir juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini berisi tentang analisis permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, lingkungan perancangan, perancangan arsitektur sistem, diagram kelas, dan struktur data.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

Bab V Pengujian dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan saran untuk pengembangan perangkat lunak selanjutnya.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang berkaitan dengan pembangunan aplikasi mobile sistem kehadiran mahasiswa yang diajukan untuk tugas akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Penelitian Terkait

Terdapat penelitian terkait rancang bangun sistem informasi pencatatan kehadiran sebelumnya, yaitu Sistem Informasi Pencatatan Kehadiran Perkuliahan Dosen dan Mahasiswa dengan Mengimplementasikan Teknologi Pemindai Sidik Jari. Penelitian ini merupakan tugas akhir yang telah selesai dikerjakan oleh salah satu mahasiswa angkatan 2011 di Jurusan Teknik Informatika ITS dengan memindai sidik jarinya pada alat pemindai sidik jari. Data sidik jari kemudian diolah dan disimpan pada *server* basis data. Sehingga kehadiran mahasiswa dan dosen dapat tersimpan dan terekap di dalam basis data. Pencatatan dengan menggunakan sidik jari memanfaatkan teknologi biometrik dimana penggunaannya lebih efektif dan efisien serta menghindari munculnya peluang dalam memanipulasi data pencatatan kehadiran serta dengan dibangunnya sistem informasi pencatatan kehadiran perkuliahan mahasiswa dan dosen lebih memudahkan dalam proses pengelolaannya. Namun sistem pencatatan kehadiran menggunakan sidik jari dirasa kurang sisi fleksibilitasnya. Fleksibilitas yang dimaksud adalah teknologi yang digunakan tidak bisa berpindah tempat dan diakses kapanpun serta harganya yang mahal.

Penelitian mengenai verifikasi wajah untuk sistem kehadiran juga pernah dilakukan sebelumnya oleh salah satu mahasiswa angkatan 2011 di Jurusan Teknik Informatika ITS. Penelitian yang dikerjakan berjudul Rancang Bangun Aplikasi Absensi

Perkuliahannya Mahasiswa dengan Pengenalan Wajah. Penelitian ini menggunakan algoritma Eigenface dari pustaka (*library*) OpenCV pada studi kasus sistem kehadiran. Implementasi algoritma ini dibangun pada aplikasi desktop dengan menggunakan kamera *webcam*. Perbedaan penelitian ini dengan tugas akhir yang akan dibuat adalah pengimplementasian algoritma yang digunakan serta penerapan *face recognition* yang akan dibangun pada *platform mobile* Android.

2.2 Teknik Informatika ITS

Jurusan Teknik Informatika yang awalnya tergabung dalam Fakultas Teknologi Industri (FTI). Mulai tahun 2002 menjadi bagian dari Fakultas Teknologi Informasi (FTIf), sesuai dengan Keputusan Menteri Pendidikan nasional RI Nomor : 109/O/2002, tanggal 2 Juli 2002 tentang Pendirian Fakultas Teknologi Informasi Pada Institut Teknologi Sepuluh Nopember.

Untuk peringkat akreditasi, Jurusan Teknik Informatika Program Studi S1 dinyatakan terakreditasi dengan nilai A berdasarkan Surat Keputusan Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) Nomor 003/BAN-PT/Ak-X/S1/V/2006, tanggal 18 Mei 2006.

Saat ini, staf pengajar jurusan teknik informatika sebanyak 48 dosen, diantaranya 6 doktor dan 23 bergelar master. Jumlah karyawan teknik informatika sebanyak 20 orang. Sedangkan mahasiswa di jurusan teknik informatika hingga tahun 2017 ini sebanyak 745 mahasiswa dengan rincian sebanyak 179 mahasiswa angkatan 2016, 178 mahasiswa angkatan 2015, 167 mahasiswa angkatan 2014, 167 mahasiswa angkatan 2013, 27 mahasiswa angkatan 2012, 14 mahasiswa angkatan 2011, 9 mahasiswa angkatan 2010 dan 2 mahasiswa angkatan 2009.

Kurikulum untuk program pendidikan S1 yang sekarang digunakan di Jurusan Teknik informatika merupakan kurikulum baru yang telah diberlakukan mulai tahun akademik 2014/2015 hingga tahun akademik 2018/2019. Dalam kurikulum ini, total Satuan Kredit Semester (SKS) normal yang harus ditempuh oleh

mahasiswa adalah 144 SKS. Beban perkuliahan sebesar 144 SKS ini dapat diselesaikan selama delapan semester. Keterangan lebih detil tentang kurikulum, dapat dilihat pada website if.its.ac.id. Total 144 SKS tersebut dikelompokkan menjadi 2 kelompok mata kuliah, yaitu: Mata Kuliah Wajib sejumlah 120 SKS (83%) dan Mata Kuliah Pilihan sejumlah 24 SKS (17%).

Jurusan Teknik Informatika menawarkan delapan bidang keahlian, yaitu komputasi cerdas dan visi (KCV); komputasi berbasis jaringan (KBJ); rekayasa perangkat lunak (RPL); algoritma dan pemrograman (AP); dasar dan terapan komputer (DTK); interaksi, grafika dan seni (IGS); manajemen informasi (MI) dan arsitektur dan jaringan komputer (AJK).

Untuk meningkatkan kenyamanan, mulai tahun 2008 Jurusan Teknik Informatika menempati gedung baru yang berjarak sekitar 500 meter ke arah timur dari gedung lama (berlokasi di dekat Grha Sepuluh Nopember ITS). Terdapat 13 ruang kelas yang semuanya dilengkapi dengan AC, LCD projector, dan wireless network. Teknik Informatika juga dilengkapi dengan perpustakaan atau ruang baca teknik informatika (RBTC). RBTC menyediakan berbagai koleksi akademik, seperti buku, majalah, jurnal akademik, laporan kerja praktik, laporan tugas akhir dan tesis. Selain itu, terdapat 13 laboratorium/ workshop yang berfungsi untuk menunjang mahasiswa dalam mengembangkan keilmuannya.[1]

2.3 Sistem Kehadiran

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.[2]

Sedangkan kehadiran menurut KBBI merupakan perihal hadir; adanya (seseorang, sekumpulan orang) pada suatu tempat.[3]

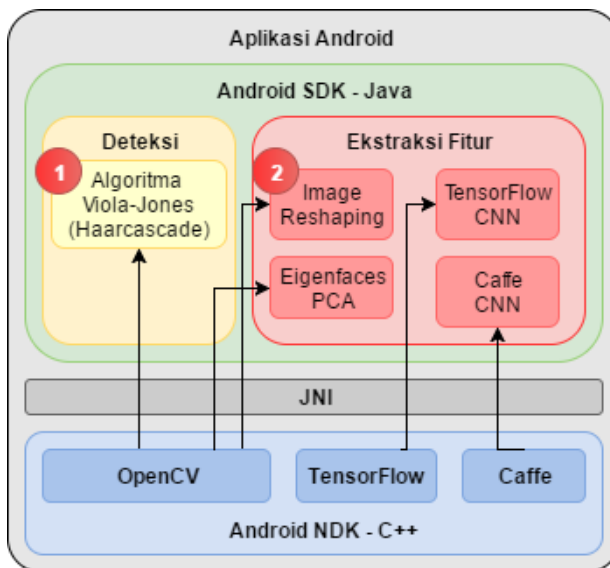
Berdasarkan definisi sistem dan kehadiran, dapat disimpulkan bahwa Sistem kehadiran adalah sistem yang digunakan dalam pengambilan data guna mengetahui jumlah kehadiran pada suatu acara. Setiap kegiatan yang membutuhkan informasi mengenai peserta tentu akan melakukan konfirmasi jika telah hadir. Hal ini juga terjadi pada proses kegiatan perkuliahan mahasiswa. Kegunaan sistem kehadiran ini terjadi pada pihak mahasiswa dan pihak dosen maupun kampus. Sistem kehadiran dapat dilakukan dengan berbagai cara, Di Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan di kertas daftar hadir mahasiswa. Salah satu kegunaan sistem kehadiran bagi mahasiswa adalah untuk syarat untuk mengikuti ujian mata kuliah, sedangkan bagi pihak dosen atau kampus, sistem kehadiran diperlukan sebagai salah satu bahan evaluasi kepada kepuasan mahasiswa terhadap suatu mata pelajaran dan pembuatan tolak ukur ke depan guna pemberian ilmu yang lebih baik.

2.4 Face Recognition

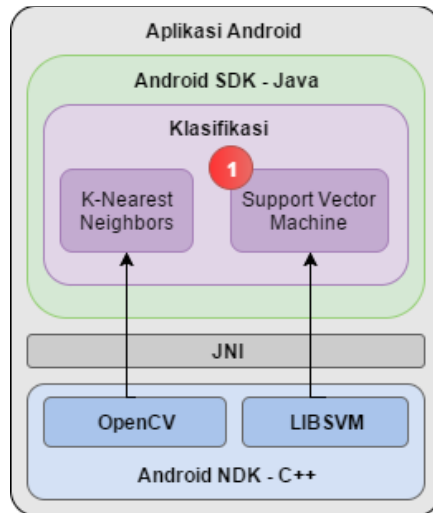
Face recognition atau pengenalan wajah adalah salah satu teknologi biometrik yang telah banyak diaplikasikan dalam sistem keamanan selain pengenalan retina mata, pengenalan sidik jari dan iris mata. Face recognition juga bagian dari aplikasi visi komputer yang mampu mengidentifikasi atau memverifikasi seseorang dari gambar digital atau frame video. Salah satu cara untuk melakukan ini adalah dengan membandingkan fitur wajah dari gambar dengan wajah yang tersimpan dalam database. Face recognition umumnya melibatkan dua tahap. Tahap pertama adalah face detection, dimana foto yang dicari untuk menemukan wajah apapun. Tahap kedua adalah face recognition, dimana wajah dideteksi dan diproses, dan dibandingkan dengan database wajah yang dikenal untuk menentukan siapa orang tersebut.[4]

2.5 Penggunaan Pustaka Android Face Recognition with Deep Learning

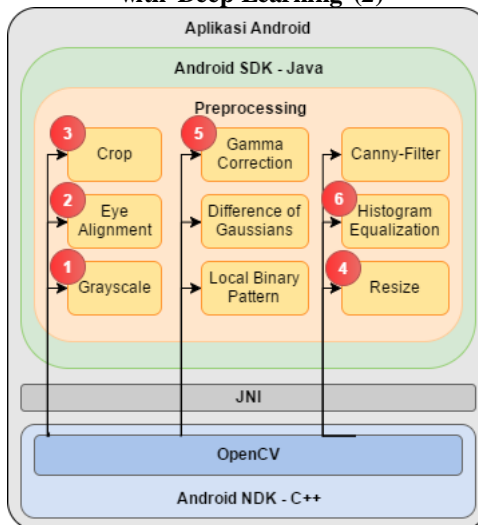
Aplikasi Mobile Android Sistem Kehadiran Mahasiswa Melalui Pencocokan Wajah ini menggunakan pustaka bantu dari project open source dari situs github.com bernama Android Face Recognition with Deep Learning Library. Pustaka ini dibuat oleh Michael Sladoje dan Mike Schachli ketika sedang menempuh skripsi di Zurich University of Applied Sciences. Hingga saat ini pustaka ini terus dikembangkan dan bersifat *open source* [5]. Pustaka ini menyediakan beberapa metode pengenalan wajah (*face recognition*) di dalamnya termasuk penggunaan pustaka LIBSVM dan OpenCV. Pada Gambar 2.1, Gambar 2.2, dan Gambar 2.3 dijabarkan arsitektur aplikasi dari pustaka ini.



Gambar 2.1 Arsitektur Aplikasi Pustaka Android Face Recognition with Deep Learning (1)



Gambar 2.2 Arsitektur Aplikasi Pustaka Android Face Recognition with Deep Learning (2)



Gambar 2.3 Arsitektur Aplikasi Pustaka Android Face Recognition with Deep Learning (3)

Aplikasi yang dibangun hanya memanfaatkan beberapa fitur dari pustaka ini. Fitur yang digunakan antara lain fitur deteksi dengan menggunakan algoritma Viola-Jones, fitur preprocessing dengan menggunakan fitur *Crop*, *Eye Alignment*, *Grayscale*, dan *Resize*, ekstraksi fitur dengan menggunakan *Image Reshaping*, dan fitur klasifikasi dengan menggunakan algoritma Support Vector Machine (SVM). Fitur-fitur ini dipilih karena memberikan hasil pencocokan wajah yang paling optimal dari penggunaan metode ini. Pada Tabel 2.1 ditunjukkan hasil dari pengujian masing-masing metode dengan menggunakan tes obyek wajah 6 orang yang berbeda dengan menggunakan 100 data training wajah yang dikumpulkan dan 100 kali tes wajah dari setiap orang. Hasil pengujian ini menjadi pertimbangan penggunaan metode ekstraksi fitur dan klasifikasi yang akan digunakan dalam tugas akhir ini.

Tabel 2.1 Hasil Pengujian Penggunaan Metode Ekstraksi Fitur dan Klasifikasi dari pustaka Android Face Recognition with Deep Learning

Metode Ekstraksi Fitur dan Klasifikasi	Akurasi	Kinerja
Eigenfaces dengan Nearest Neighbour	85.29%	0.46 detik / gambar
Image Reshaping dengan SVM	95.29%	0.58 detik / gambar
TensorFlow Inception model dengan SVM	88.57%	2.35 detik / gambar
TensorFlow VGG Face Descriptor Model dengan SVM	100%	6.27 detik / gambar

2.5.1 Penggunaan OpenCV Library

OpenCV atau Open Source Computer Vision Library adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara *real-time*, OpenCV dibuat oleh Intel. Program ini bebas dan berada dalam naungan open source dari lisensi BSD. Pustaka ini merupakan pustaka *cross-platform*. Program ini

didedikasikan sebagian besar untuk pengolahan citra secara *real-time*[6].

Dalam pustaka Android Face Recognition with Deep Learning, pustaka (*library*) OpenCV diimplementasikan pada proses deteksi wajah, proses ekstraksi fitur, *preprocessing*, proses pengolahan foto wajah terdeteksi, serta fitur-fitur bantuan yang digunakan untuk pengolahan citra dinamis. Pada proses deteksi wajah digunakan algoritma Viola-Jones atau Haarcascade yang ditunjukkan oleh nomor 1 pada Gambar 2.1. Pada proses ekstraksi fitur akan digunakan algoritma *image reshaping* yang ditunjukkan oleh nomor 2 pada Gambar 2.1. Untuk tahap *preprocessing* akan digunakan beberapa tahapan antara lain *Standard Preprocessing*, *Brightness Correction*, *Contrast Adjustment*, dan *Standard Postprocessing*. Algoritma dari tahap *Standard Preprocessing* yang digunakan ditunjukkan pada nomor 1, 2, dan 3 pada Gambar 2.3 yaitu *Grayscale*, *Crop*, dan *Eye Alignment*. Pada tahap *Brightness Correctness* dan *Contrast Adjustment* digunakan algoritma Gamma Correction dan Histogram Equalization. Algoritma ini ditunjukkan pada nomor 5 dan 6 pada Gambar 2.3. Pada tahap *Standard Postprocessing* digunakan algoritma *Resize* untuk meningkatkan performa dari pengecilan gambar. Algoritma tahap ini ditunjukkan pada nomor 4 Gambar 2.3.

2.5.1.1 Penggunaan Algoritma Tahap Deteksi Wajah

Pada tahap pendeteksian wajah digunakan algoritma Haarcascade atau dikenal dengan Viola-Jones. Algoritma Haarcascade adalah metode pendeteksian objek yang efektif untuk digunakan. Algoritma ini diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001 dalam penelitian mereka yang berjudul “Rapid Object Detection using a Boosted Cascade of Simple Features”. Algoritma ini merupakan sebuah pendekatan berbasis *machine learning* dimana fungsi *cascade* dilatih dari banyak gambar positif (gambar wajah) dan gambar negatif (gambar tanpa wajah) yang kemudian hasilnya digunakan untuk mendeteksi objek pada gambar lainnya [7].

Dalam pustaka (*library*) OpenCV, algoritma ini sudah dilengkapi dengan banyak *pre-trained classifiers* untuk wajah, mata, senyuman, dan lain-lain. *Classifiers* ini disimpan dalam bentuk *file XML*. Untuk implementasi dari algoritma ini sudah diterapkan oleh pustaka (*library*) Android Face Recognition with Deep Learning, sehingga dalam pembuatan aplikasi tugas akhir ini hanya tinggal memakai saja.

2.5.1.2 Penggunaan Algoritma Tahap *Preprocessing*

Tahap *preprocessing* yang dilakukan dalam pustaka (*library*) Android Face Recognition with Deep Learning meliputi beberapa tahapan. Tahap-tahap yang dilakukan antara lain:

1. *Standard Preprocessing*
2. *Brightness Correction*
3. *Contours*
4. *Contrast Adjustment*
5. *Standard Postprocessing*

Pada tahap *Standard Preprocessing* dilakukan proses *Grayscale*, *Crop*, dan *Eye Alignment*. Proses *Grayscale* adalah proses mengubah warna gambar menjadi berwarna abu-abu. Proses *Crop* adalah proses untuk memotong gambar tepat pada area wajah, dan *Eye Alignment* adalah proses untuk mengubah posisi gambar supaya posisi mata sejajar secara horizontal.

Pada tahap *Brightness Correction* dilakukan proses koreksi kecerahan dari gambar yang terdeteksi oleh kamera. Tahap *Contours* diperlukan untuk menambah keakuratan untuk analisis bentuk dan pengenalan objek wajah. *Contours* merepresentasikan sebuah kurva yang menggabungkan semua titik-titik kontinyu yang memiliki warna atau intensitas yang sama. Pada tahap *Contrast Adjustment* dilakukan pengaturan kontras warna. Pada tahap *Standard Postprocessing* dilakukan proses *Resize*. Proses *Resize* bertujuan untuk memperkecil ukuran gambar wajah yang sudah diolah dari tahap 1 sampai 4. Pengubahan ukuran gambar wajah ini dimaksudkan untuk meningkatkan performa dari proses ekstraksi fitur.

2.5.1.3 Penggunaan Algoritma Tahap Ekstraksi Fitur

Dalam *image processing*, ekstraksi fitur dimulai dari set awal data yang terukur. Ekstraksi fitur bertujuan membangun nilai-nilai turunan (fitur) hingga menjadi nilai yang informatif dan *non-redundant* (tidak berulang), memfasilitasi langkah pembelajaran dan generalisasi selanjutnya, dan dalam kasus tertentu dapat memberikan hasil interpretasi manusia yang lebih baik [8]. Ekstraksi fitur biasanya berkaitan dengan dengan pengurangan dimensi dari data.

Pada implementasi tahap ekstraksi fitur dari aplikasi ini akan digunakan algoritma *Image Reshaping* yang telah tersedia di dalam pustaka (*library*) Android Face Recognition with Deep Learning. Algoritma ini akan mengubah dimensi nilai matriks dari gambar wajah yang sudah diambil dan diolah pada tahap *preprocessing* menjadi matriks 1 dimensi saja karena gambar hanya berupa foto wajah. Hasil dari algoritma ini kemudian dipakai untuk tahap klasifikasi.

2.5.2 Penggunaan LIBSVM

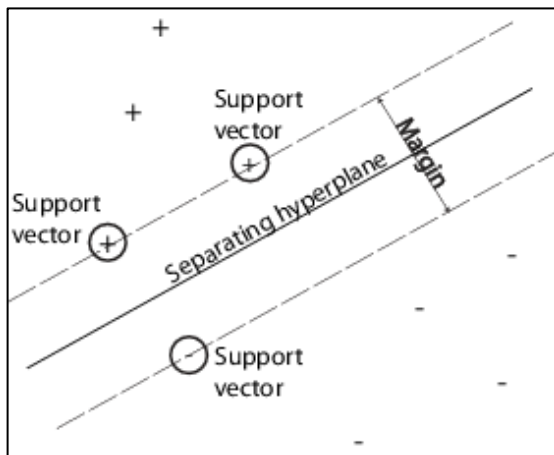
LIBSVM merupakan sebuah pustaka untuk Support Vector Machines (SVM). Pustaka ini telah dikembangkan sejak tahun 2000. Tujuan dari pembuatan pustaka ini adalah untuk mempermudah pengguna untuk menerapkan SVM untuk aplikasi yang dibuat oleh pengguna tersebut. Pustaka ini sudah banyak digunakan untuk berbagai macam *machine learning* dan studi kasus lainnya [9].

Dalam pustaka Android Face Recognition with Deep Learning, pustaka (*library*) LIBSVM digunakan pada saat proses klasifikasi data foto yang terkumpul. Penggunaan pustaka ini ditunjukkan pada nomor 1 Gambar 2.2. Tipe kernel yang dipakai menggunakan pengaturan awal (*default*) dari pustaka Android Face Recognition with Deep Learning. Tipe kernel yang dipakai adalah linear kernel. Tipe ini digunakan karena gambar memiliki dimensi yang cukup untuk memberikan akurasi yang baik.

2.5.2.1 Penggunaan Algoritma Tahap Klasifikasi

Support Vector Machine adalah model pembelajaran mesin yang umumnya digunakan untuk mengatasi masalah klasifikasi dan regresi. Klasifikasi dilakukan dengan cara menemukan hyperplane yang optimal (garis optimal untuk data 2 dimensi) yang memisahkan kelas data dengan margin maksimum seperti pada Gambar 2.4. Ketika nilai margin lebih besar umumnya hasil klasifikasi lebih baik generalisasi. Data pelatihan (*training data*) dengan menggunakan algoritma SVM menghasilkan vektor pendukung (*support vector*). Vektor pendukung adalah data titik yang terletak paling dekat dengan hyperplane tersebut, dan fungsi keputusan SVM sepenuhnya bergantung pada nilai dari vektor pendukung ini.

Pada tahap klasifikasi digunakan algoritma Support Vector Machine yang telah disiapkan dalam pustaka (*library*) Android Face Recognition with Deep Learning. Algoritma SVM yang digunakan menggunakan kernel linear karena fitur-fitur yang dihasilkan dari gambar foto wajah sederhana.



Gambar 2.4 Hyperplane yang optimal memisahkan dua kelas (+) dan (-) dalam SVM

2.6 API

API atau Application Programming Interface adalah sekumpulan perintah, fungsi, serta *protocol* yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan *programmer* untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API juga dapat menjelaskan cara sebuah tugas tertentu dilakukan. Sebuah API juga dapat digunakan untuk menspesifikasikan cara komponen aplikasi saling berinteraksi. Dengan bahasa yang lebih sederhana, API adalah fungsi-fungsi pemrograman yang disediakan oleh aplikasi atau layanan agar layanan tersebut bisa diintegrasikan dengan aplikasi yang dibuat. Di dalam pemrograman Android, API biasanya digunakan untuk mempermudah pertukaran data dari atau ke dalam server [10].

2.7 JSON

JSON atau JavaScript Object Notation adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format JSON dibuat berdasarkan bagian dari bahasa pemrograman JavaScript [11].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas analisis kebutuhan dan rancangan yang akan digunakan untuk membangun perangkat lunak yang diajukan sebagai tugas akhir

3.1 Analisis

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

3.1.1 Analisis Permasalahan

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Sistem kehadiran ini dapat menjadi bukti bahwa seorang mahasiswa telah berada di dalam kelas. Pencatatan kehadiran mahasiswa dapat dilakukan dengan berbagai cara. Di Jurusan Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran. Kecurangan ini sering disebut titip absen. Titip absen merupakan kejadian di mana mahasiswa tercatat hadir pada kertas daftar kehadiran kelas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Dengan adanya celah kelemahan ini, mahasiswa lain dapat melakukan tanda tangan untuk mahasiswa lain. Oleh karena itu, diperlukan suatu sistem yang dapat digunakan untuk mengatasi kecurangan dalam pencatatan kehadiran mahasiswa di dalam kelas.

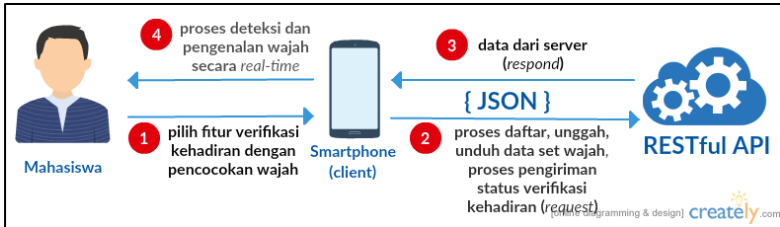
Perkembangan teknologi informasi di dunia semakin cepat, khususnya pada teknologi perangkat bergerak (*mobile*). Akses dan pengambilan informasi juga semakin mudah dan cepat diakses melalui penggunaan perangkat bergerak maupun web. Perkembangan ini juga mempengaruhi cara seseorang dalam melakukan proses verifikasi untuk masuk ke dalam sebuah sistem.

Verifikasi merupakan sebuah proses yang wajib ada dimana diperlukan sebuah kebenaran data/informasi misalnya verifikasi keaslian identitas. Proses verifikasi ini sudah banyak diterapkan seiring dengan perkembangan teknologi informasi oleh perusahaan, birokrasi, dan lain-lain. Proses verifikasi identitas yang sering digunakan adalah menggunakan alat deteksi sidik jari (*fingerprint*). Namun kelemahan dari penerapan ini adalah kurangnya fleksibilitas dari proses verifikasi ini. Fleksibilitas yang dimaksud adalah teknologi yang digunakan tidak bisa berpindah tempat dan diakses kapanpun. Oleh karena itu, tugas akhir ini akan mengimplementasikan sebuah aplikasi mobile sistem kehadiran berbasis Android. Implementasi pada aplikasi mobile diharapkan mampu menambah fleksibilitas dari sistem kehadiran yang sudah ada saat ini. Aplikasi kehadiran yang akan dibuat akan dibatasi ruang lingkup penggunaannya. Pengguna dari aplikasi ini adalah mahasiswa Teknik Informatika ITS.

Sistem kehadiran pada aplikasi ini mengimplementasikan metode pencocokan wajah (*face recognition*) ke dalam aplikasi mobile. Aplikasi ini diimplementasikan dengan menggunakan pustaka Android Face Recognition with Deep Learning dimana di dalamnya terdapat algoritma-algoritma dari pustaka (*library*) OpenCV versi 3.2.0 dan pustaka (*library*) LIBSVM. Secara umum, Prosedur pencocokan wajah dibagi menjadi dua tahapan. Tahap pertama adalah deteksi wajah (*face detection*), yaitu tahap di mana aplikasi mencari wajah dalam gambar tangkapan kamera untuk memastikan adanya wajah seseorang yang tertangkap. Tahap kedua adalah tahap identifikasi atau pengenalan wajah (*face recognition*). Pada tahap ini dilakukan perbandingan hasil ekstraksi fitur wajah yang terdeteksi dengan gambar wajah yang tersimpan dalam *database*. Perbandingan ini dilakukan supaya sistem mendapatkan hasil identifikasi identitas seseorang dari wajah yang terdeteksi. Pemberian identitas ini dilakukan dengan melakukan *training* terhadap aplikasi sebelumnya. Langkah *training* melibatkan pendeteksian wajah yang kemudian hasil deteksi ini diolah dengan menggunakan beberapa teknik pemrosesan gambar supaya dapat

diterima dengan baik oleh aplikasi, setelah itu mengajarkan sistem siapa identitas wajah yang terdeteksi dengan menggunakan algoritma *training*.

3.1.2 Deskripsi Umum Sistem



Gambar 3.1 Deskripsi Umum Sistem

Gambar 3.1 merupakan arsitektur sistem perangkat lunak yang akan dibangun. Pada gambar ditunjukkan fokus pengerjaan pada pada perangkat lunak yang akan dibangun. Sistem yang dibangun meliputi pendeteksian wajah dari kamera *client/smartphone*, proses pencocokan wajah di sisi *client/smartphone*, hingga pengiriman status verifikasi kehadiran ke *database server*. Pada gambar ini ditunjukkan tahapan *input*, proses, dan *output* pada fokus pengerjaan tugas akhir ini. Tahapannya adalah sebagai berikut.

1. Tahap *input* aplikasi mobile sistem kehadiran mahasiswa ditunjukkan pada nomor 1. Input aplikasi yang diberikan adalah pemilihan fitur verifikasi kehadiran dengan pencocokan wajah oleh mahasiswa, pemilihan fitur kelola data wajah mahasiswa, dan pemilihan fitur verifikasi kehadiran.
2. Tahap proses ditunjukkan pada nomor 2, 3, dan 4. Pada tahap ini, aplikasi mobile sistem kehadiran mahasiswa melakukan proses pendaftaran data wajah dari *client* ke sistem server, proses unggah dan unduh data wajah dari dan ke *database server* dengan penyimpanan internal *smartphone*, proses *training* data set wajah, proses *face*

detection dan *face recognition*, serta proses pengiriman status verifikasi kehadiran mahasiswa.

3. Tahap *output* aplikasi mobile sistem kehadiran mahasiswa ditunjukkan pada nomor 2. *Output* dari aplikasi mobile sistem kehadiran mahasiswa ini adalah pengiriman status mahasiswa berhasil melakukan verifikasi kehadiran ke sistem server.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan cakupan perangkat lunak yang telah dijelaskan pada bagian deskripsi umum sistem, dibutuhkan adanya spesifikasi perangkat lunak supaya dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Diharapkan dengan adanya spesifikasi ini dapat menyesuaikan kebutuhan-kebutuhan pengguna. Spesifikasi kebutuhan perangkat lunak adalah penjelasan mengenai kebutuhan sistem yang diinginkan pengguna atau *client* dalam bentuk tulisan. Spesifikasi kebutuhan perangkat lunak pada tugas akhir ini terdiri dari kebutuhan fungsional, kebutuhan non-fungsional, aktor, dan kasus penggunaan.

3.1.3.1 Kebutuhan Fungsional

Tabel 3.1 Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Mendaftarkan data wajah untuk proses pencocokan wajah.	Melakukan proses pendaftaran data wajah.
2	Melakukan presensi dengan pencocokan wajah.	Melakukan proses presensi melalui pencocokan wajah.

3.1.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem adalah sebagai berikut.

1. Kebutuhan Performa
Perangkat lunak akan berjalan dengan performa terbaik apabila dijalankan di atas spesifikasi perangkat minimal.
2. Kebutuhan Perlindungan Keamanan
Perangkat lunak dapat memverifikasi kehadiran mahasiswa sesuai dengan pengguna yang memiliki hak.

3.1.3.3 Aktor

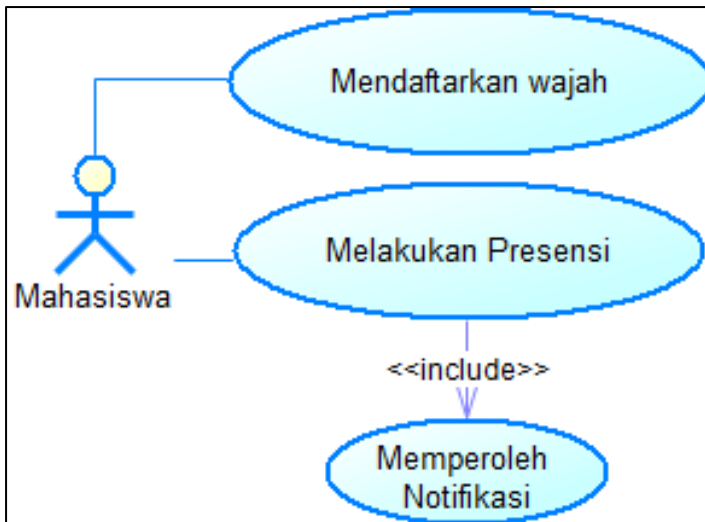
Pengertian aktor atau biasa disebut entitas luar adalah siapa saja atau pihak-pihak apapun yang harus berinteraksi dengan sistem. Aktor dapat berupa manusia, sistem lain, atau perangkat lain. Pada perangkat lunak ini terdapat satu pengguna yaitu mahasiswa. Mahasiswa adalah sebutan bagi orang yang sedang menempuh pendidikan tinggi di sebuah perguruan tinggi yang terdiri atas sekolah tinggi, akademi, atau universitas.

3.1.3.4 Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan digambarkan dalam tabel penjelasan kasus penggunaan dan diagram kasus penggunaan. Tabel penjelasan kasus penggunaan dapat dilihat pada Tabel 3.2 dan diagram kasus penggunaan dapat dilihat pada Gambar 3.2.

Tabel 3.2 Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Aktor
UC-001	Mendaftarkan wajah	Mahasiswa
UC-002	Melakukan presensi	Mahasiswa



Gambar 3.2 Diagram Kasus Penggunaan

3.1.3.4.1 Mendaftarkan Wajah (UC-001)

Kasus penggunaan nomor UC-001 ini diakses ketika mahasiswa hendak mendaftarkan foto wajahnya ke dalam sistem sebagai data pencocokan wajah yang dipakai pada saat proses presensi. Spesifikasi, diagram aktivitas, diagram sekuens, dan diagram kolaborasi objek dari kasus penggunaan ini dapat dilihat pada Tabel 3.3,

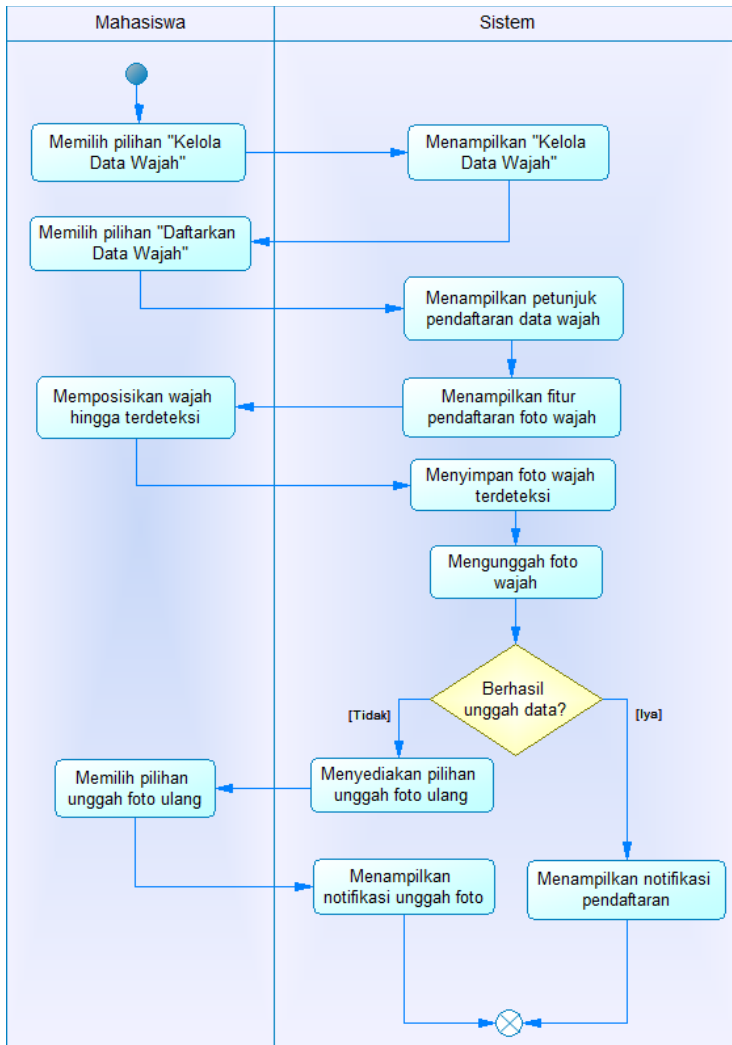
Gambar 3.3, Gambar 3.4, dan Gambar 3.5.

Tabel 3.3 Spesifikasi Kasus Penggunaan UC-001

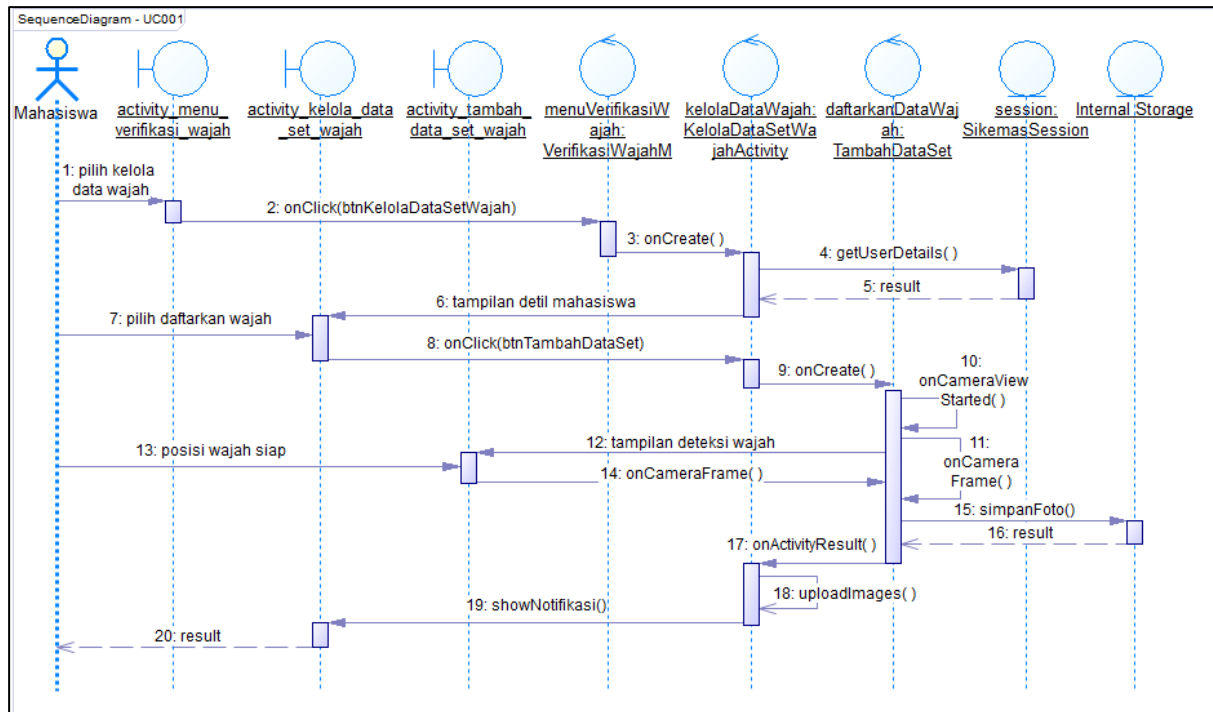
Kode Use Case	UC-001
Nama Use Case	Mendaftarkan wajah
Aktor	Mahasiswa
Deskripsi	Mahasiswa dapat mendaftarkan foto wajahnya sebagai data wajah ke sistem pada sisi server

Relasi	-	
Kondisi Awal	Sistem belum memiliki data wajah mahasiswa	
Kondisi Akhir	Sistem sudah memiliki data wajah mahasiswa	
Alur Kejadian Normal	Aktor	Sistem
	1. Memilih pilihan “Kelola Data Wajah”	
		2. Menampilkan “Kelola Data Wajah”
	3. Memilih pilihan “Daftarkan Data Wajah”	
		4. Menampilkan petunjuk pendaftaran data wajah
		5. Menampilkan fitur pendaftaran foto wajah
	6. Memposisikan wajah hingga terdeteksi	
		7. Menyimpan foto wajah terdeteksi
		8.a. Mengunggah foto wajah
		9. Menampilkan notifikasi pendaftaran
Alur Kejadian Alternatif	Aktor	Sistem
		8.b. Menyediakan pilihan unggah data wajah
	8.b.1. Memilih pilihan unggah data wajah	
		8.b.2 Menampilkan notifikasi unggah data wajah

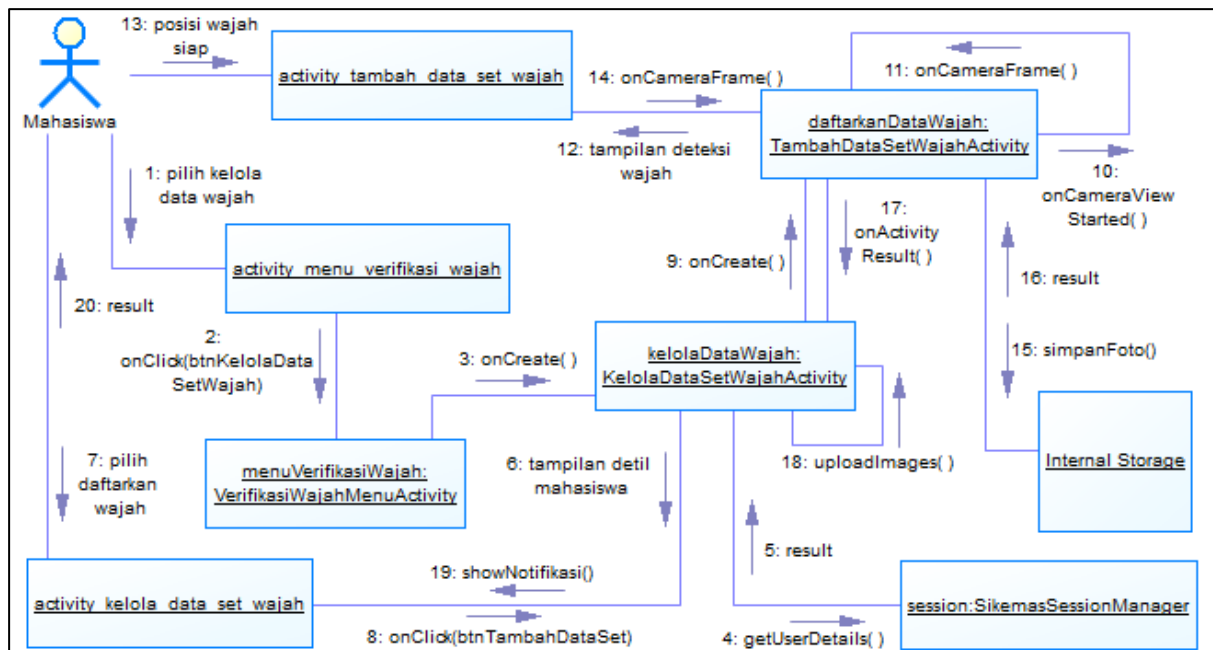
Alur kejadian alternatif ini dilakukan ketika terjadi kesalahan dalam proses unggah foto ke sistem server.



Gambar 3.3 Diagram Aktivitas UC-001



Gambar 3.4 Diagram Sekuens UC-001



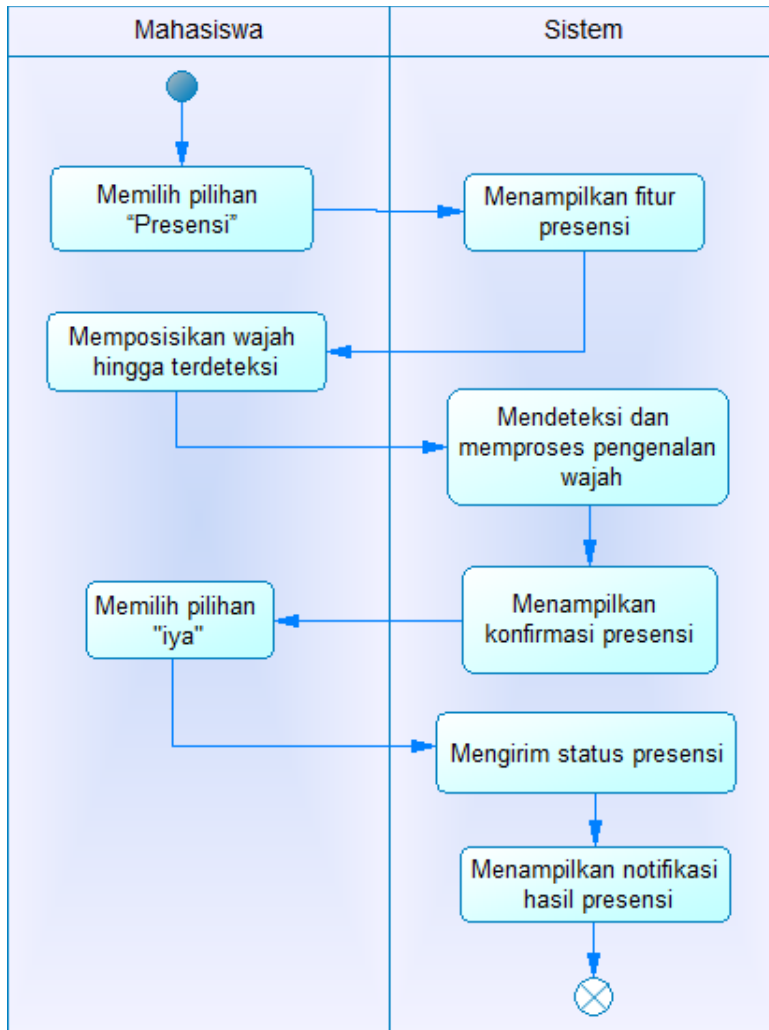
Gambar 3.5 Diagram Kolaborasi Objek UC-001

3.1.3.4.2 Melakukan Presensi (UC-002)

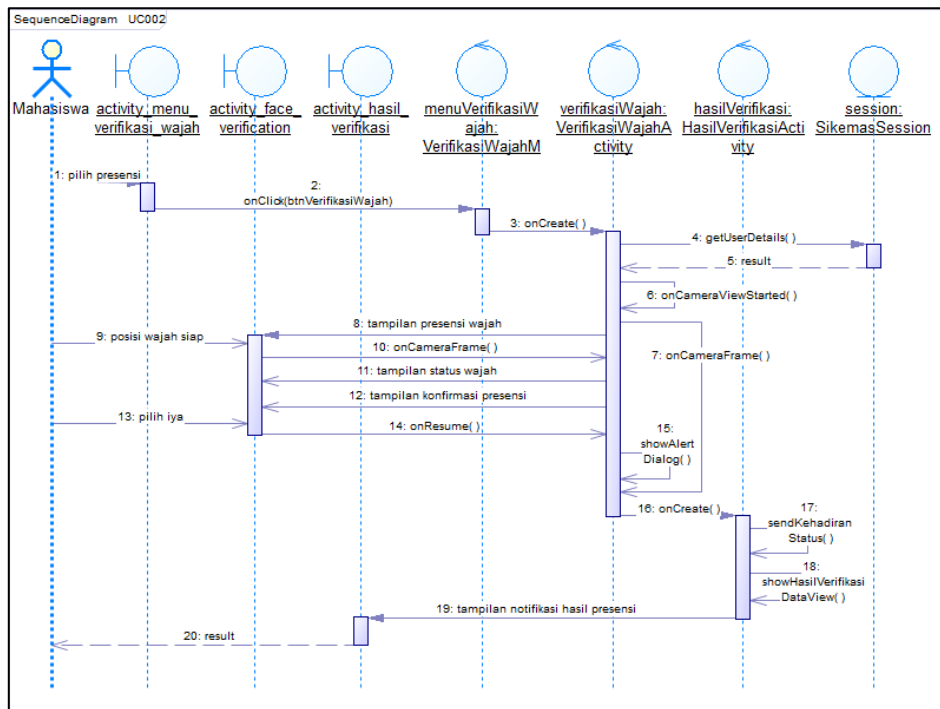
Kasus penggunaan nomor UC-002 ini diakses ketika mahasiswa hendak melakukan presensi melalui fitur pencocokan wajah. Spesifikasi, diagram aktivitas, diagram sekuens, dan diagram kolaborasi objek dari kasus penggunaan ini dapat dilihat pada Tabel 3.4, Gambar 3.6, Gambar 3.7, Gambar 3.8.

Tabel 3.4 Spesifikasi Kasus Penggunaan UC-002

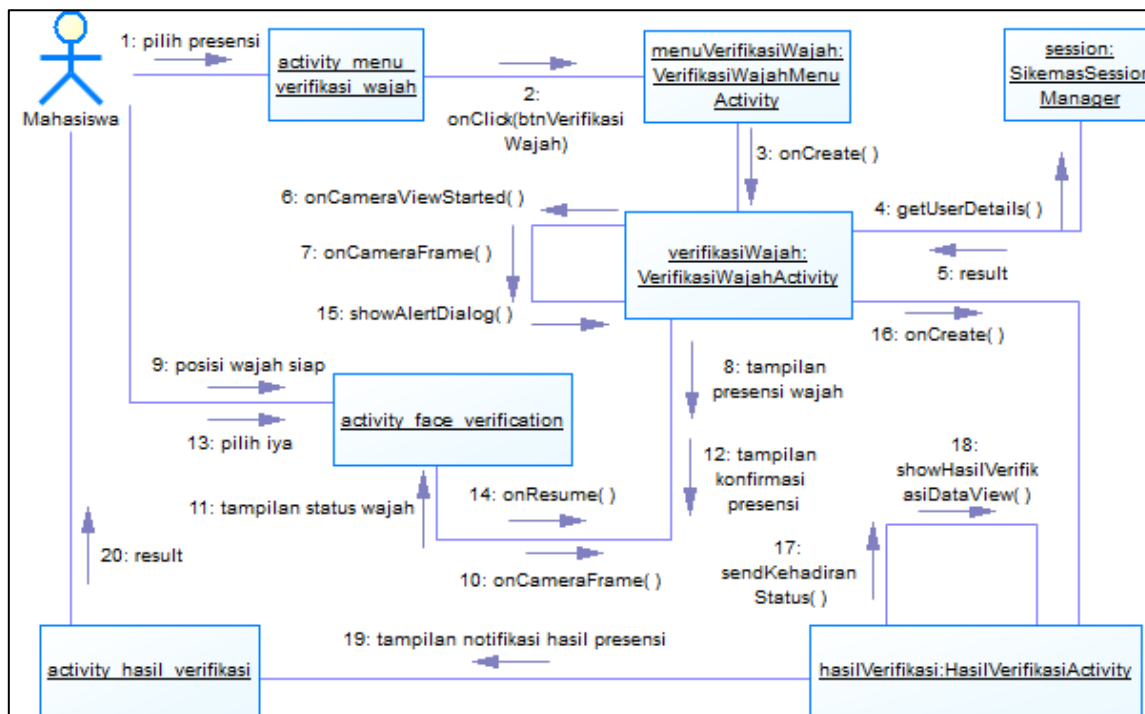
Kode Use Case	UC-002	
Nama Use Case	Melakukan presensi	
Aktor	Mahasiswa	
Deskripsi	Mahasiswa dapat melakukan presensi melalui pencocokan wajah	
Relasi	-	
Kondisi Awal	Sistem sudah memiliki data wajah dan data hasil latihan (<i>training</i>) data wajah.	
Kondisi Akhir	Sistem menerima status presensi	
Alur Kejadian Normal	Aktor	Sistem
	1. Memilih pilihan “Presensi”.	
		2. Menampilkan fitur presensi
	3. Memposisikan wajah hingga terdeteksi.	
		4. Mendeteksi wajah dan memproses pengenalan wajah
		5. Menampilkan konfirmasi presensi
	6. Memilih pilihan “iya”	
		7. Mengirimkan status presensi
		8. Menampilkan notifikasi hasil presensi



Gambar 3.6 Diagram Aktivitas UC-002



Gambar 3.7 Diagram Sekuens UC-002



Gambar 3.8 Diagram Kolaborasi Objek UC-002

3.2 Perancangan

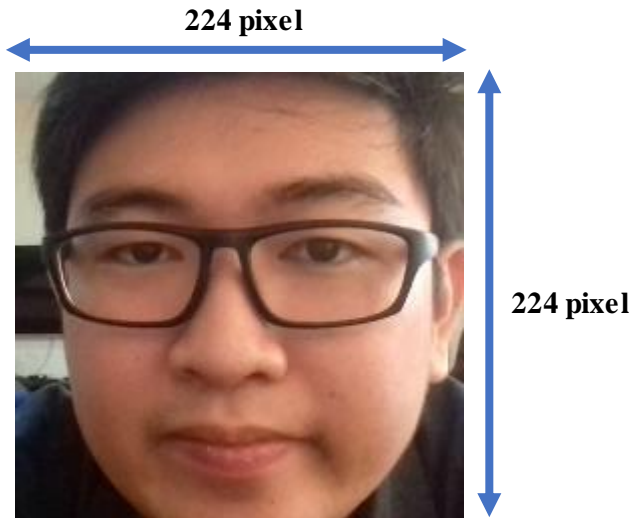
Perancangan dalam subbab ini membahas perancangan dari aplikasi tugas akhir bernama Sikemas. Subbab ini terdiri dari perancangan data, perancangan arsitektur perangkat lunak, perancangan diagram kelas, perancangan antarmuka pengguna, dan perancangan proses.

3.2.1 Perancangan Data

Perancangan data dari aplikasi Sikemas dibuat untuk memberi penjelasan berkaitan dengan data yang dipakai dan dihasilkan oleh aplikasi ini. Perancangan data yang dilakukan meliputi penentuan jenis file yang digunakan, ekstensi file yang bisa dipakai, ukuran file, serta lokasi penyimpanan dari file yang dipakai dan dihasilkan oleh aplikasi ini.

Jenis data yang digunakan adalah data gambar atau *image*. Jenis data ini digunakan untuk menyimpan hasil dari pengambilan data wajah pada kasus pengguna mendaftarkan wajah. Jenis file ini juga digunakan untuk penyimpanan hasil unduhan data wajah. Semua data gambar atau *image* ini disimpan dengan ekstensi PNG. Ukuran dari data gambar atau *image* ini ditentukan untuk ukuran dimensi gambar 224 x 224 pixel. Gambar 3.9 merupakan contoh ukuran data *image* yang sudah dicrop pada bagian wajah dan memiliki ukuran dimensi 224 x 224 pixel sesuai dengan perancangan ukuran dimensi data *image*.

Penyimpanan untuk semua data dari aplikasi Sikemas akan disimpan di penyimpanan internal *smartphone*. Lokasi penyimpanan data Sikemas berada di *folder* facerecognition. Data *image* yang diambil pada saat pendaftaran wajah atau saat pengunduhan data wajah akan disimpan pada *subfolder* facerecognition bernama training. Data *image* hasil dari pelatihan (*training*) data wajah dan file teks hasil dari klasifikasi algoritma SVM akan disimpan di *subfolder* dari facerecognition bernama data.



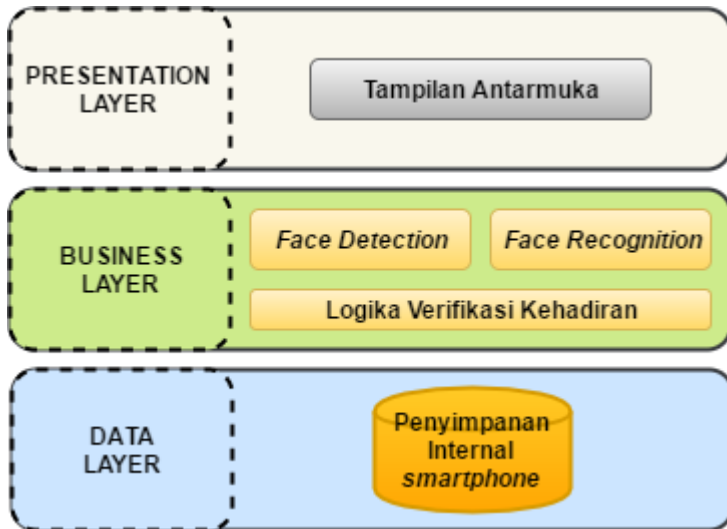
Gambar 3.9 Contoh Gambar Wajah Terambil Dan Disimpan

3.2.2 Perancangan Arsitektur Perangkat Lunak

Arsitektur sistem untuk Tugas Akhir ini menggunakan arsitektur 3 lapis yang terdiri dari *Data Layer*, *Business Layer*, dan *Presentation Layer*. *Data Layer* berfungsi untuk menangani basis data aplikasi serta penyimpanan internal aplikasi, *Business Layer* berfungsi untuk menangani logika aplikasi serta proses-proses verifikasi kehadiran mahasiswa. Sedangkan *Presentation Layer* berfungsi menangani tampilan antarmuka pengguna. *Data Layer* pada aplikasi ini ditangani oleh penyimpanan internal *smartphone* yang diatur oleh *File system* dari sistem operasi Android dan *shared preference* untuk menyimpan data sama yang dibutuhkan oleh aplikasi. Gambar arsitektur aplikasi sistem yang dibangun dapat dilihat pada Gambar 3.10.

Pada lapisan *Business Layer*, digunakan RESTful API *web service* sebagai jembatan komunikasi atau pertukaran data antara sistem *client* dengan sistem *server*. Lapisan *Business Layer*

melakukan pengolahan API ini supaya dapat dipakai oleh aplikasi. API yang diolah oleh aplikasi akan menggunakan tipe JSON. Pengolahan API ini dapat dilihat pada hasil semua implementasi kasus penggunaan.



Gambar 3.10 Diagram Arsitektur Sistem

3.2.3 Perancangan Diagram Kelas

Perancangan diagram kelas dapat dilihat pada Gambar 3.21. Pada diagram kelas ini akan ditunjukkan kelas-kelas beserta beberapa atribut dan fungsi penting yang akan digunakan dalam aplikasi Sikemas.

3.2.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan dan tampilan yang menarik bagi penggunanya. Sesuai dengan gambaran umum perangkat lunak

yang dikerjakan dalam aplikasi Sikemas, antarmuka pengguna hanya dirancang untuk satu hak akses pengguna, yaitu mahasiswa. Pada hak akses mahasiswa, terdapat halaman menu utama verifikasi wajah, kelola data wajah, mendaftarkan data wajah, progress laithan data wajah, dan melakukan presensi.

3.2.4.1 Rancangan Halaman Antarmuka Menu Utama Verifikasi Wajah

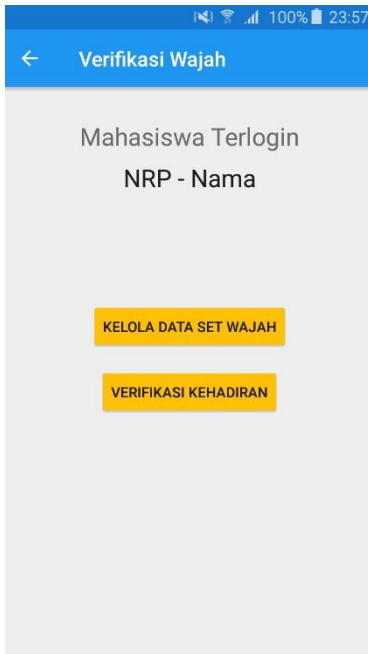
Halaman ini akan ditampilkan pada aplikasi ketika mahasiswa memilih pilihan pencocokan wajah pada tampilan sistem yang lain. Rancangan dapat dilihat pada Gambar 3.11. Atribut antarmuka dapat dilihat pada Tabel 3.5.

Tabel 3.5 Atribut Antarmuka Menu Utama Verifikasi Wajah

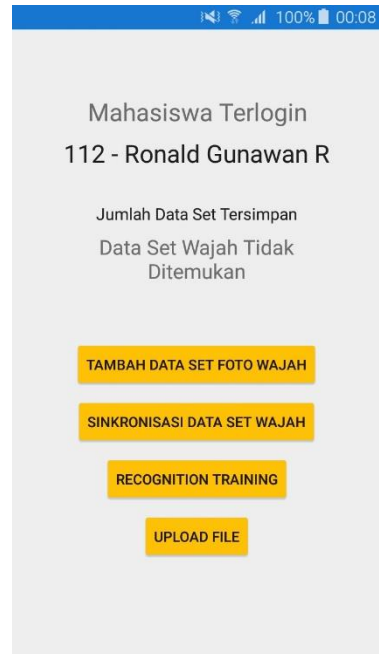
No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>tv_user_detail_info</i>	<i>TextView</i>	Informasi nomor induk (nrp/nip) dan nama mahasiswa yang masuk dalam sistem	<i>String</i>
2	<i>btn_kelola_data_set_wajah</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman antarmuka kelola data set wajah	<i>ButtonClick</i>
3	<i>btn_verifikasi_view</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman antarmuka verifikasi kehadiran	<i>ButtonClick</i>

3.2.4.2 Rancangan Halaman Antarmuka Kelola Data Wajah

Halaman ini ditampilkan pada aplikasi ketika mahasiswa memilih pilihan kelola data wajah pada tampilan antarmuka menu utama verifikasi wajah. Rancangan halaman dapat dilihat pada Gambar 3.12. Atribut antarmuka dapat dilihat pada Tabel 3.6.



Gambar 3.11 Rancangan Halaman Antarmuka Menu Utama Verifikasi Wajah



Gambar 3.12 Rancangan Halaman Antarmuka Kelola Data Wajah

Tabel 3.6 Atribut Antarmuka Kelola Data Wajah

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>tv_user_detail_info</i>	<i>TextView</i>	Informasi nomor induk (nrp/nip) dan nama mahasiswa yang masuk dalam sistem	<i>String</i>
2	<i>tv_data_set</i>	<i>TextView</i>	Informasi jumlah data set wajah yang ditemukan dalam perangkat mobile mahasiswa	<i>String</i>
3	<i>btn_tambah_data_set_wajah</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman antarmuka tambah data set wajah	<i>ButtonClick</i>
4	<i>btn_sinkronisasi_data_set_wajah</i>	<i>Button</i>	Tombol aksi untuk melakukan sinkronisasi data set wajah dari sistem server ke perangkat mobile mahasiswa	<i>ButtonClick</i>
5	<i>btn_recognition_training</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman	<i>ButtonClick</i>

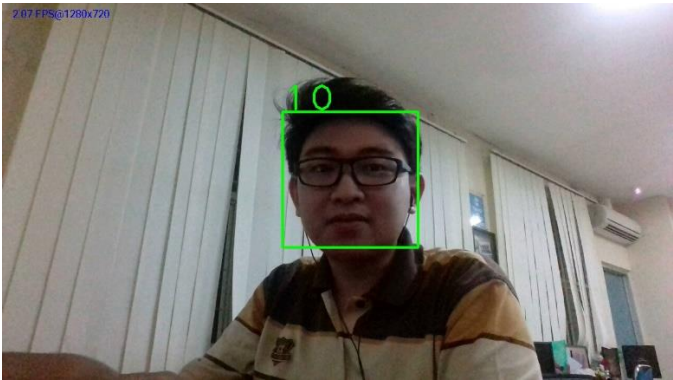
No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
			antarmuka training data set wajah	
6	<i>btn_upload_file</i>	<i>Button</i>	Tombol aksi untuk melakukan pengunggahan data dari perangkat mobile mahasiswa ke sistem server	<i>ButtonClick</i>

3.2.4.3 Rancangan Halaman Antarmuka Mendaftarkan Data Wajah

Halaman ini akan ditampilkan sistem ketika mahasiswa memilih pilihan daftarkan data wajah pada halaman antarmuka kelola data wajah. Rancangan dapat dilihat pada Gambar 3.13 Atribut antarmuka dapat dilihat pada Tabel 3.7.

Tabel 3.7 Atribut Antarmuka Tambah Data Wajah

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>tambah_set_wajah_previ</i> <i>ew</i>	<i>CustomCameraView</i> (dari pustaka Android Face Recognition)	Tampilan kamera dari pustaka Android Face Recognition untuk mengambil foto wajah mahasiswa	<i>InputStream</i> , <i>File Image</i>



Gambar 3.13 Rancangan Halaman Antarmuka Tambah Data Wajah

3.2.4.4 Rancangan Halaman Antarmuka Melatih Data Wajah

Halaman ini akan ditampilkan ketika mahasiswa menekan pilihan latih data wajah pada halaman antarmuka kelola data wajah. Halaman ini akan memuat hasil pencatatan progress latihan data wajah yang sedang dilakukan oleh aplikasi. Rancangan dapat dilihat pada Gambar 3.15. Atribut antarmuka dapat dilihat pada Tabel 3.8.

Tabel 3.8 Atribut Antarmuka Latih Data Wajah

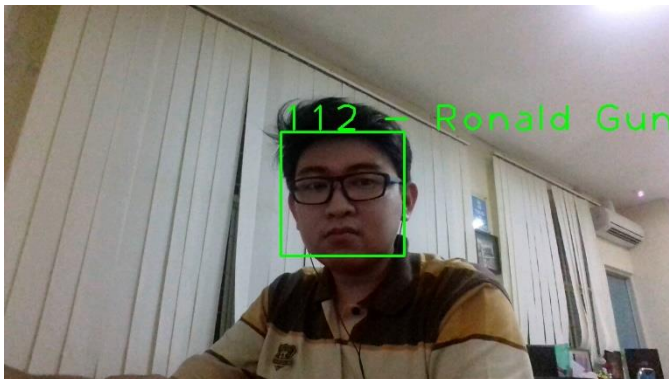
No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>tv_training_progress</i>	<i>TextView</i>	Informasi keterangan progress data set yang sedang dilatih	<i>String</i>

3.2.4.5 Rancangan Halaman Antarmuka Melakukan Presensi

Halaman ini akan ditampilkan ketika mahasiswa menekan pilihan presensi pada halaman antarmuka menu utama verifikasi wajah. Halaman ini akan memuat hasil pencatatan progress latihan wajah yang sedang dilakukan oleh aplikasi. Rancangan dapat dilihat pada Gambar 3.14. Atribut antarmuka dapat dilihat pada Tabel 3.9.

Tabel 3.9 Atribut Antarmuka Melakukan Presensi

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>tv_user_detail</i>	<i>CustomCameraView</i> (dari pustaka Android Face Recognition)	Informasi nomer induk (nrp/nip) dan nama mahasiswa yang masuk dalam sistem	<i>String</i>



Gambar 3.14 Rancangan Halaman Antarmuka Melakukan Presensi



Gambar 3.15 Rancangan Halaman Antarmuka Training Data Set Wajah

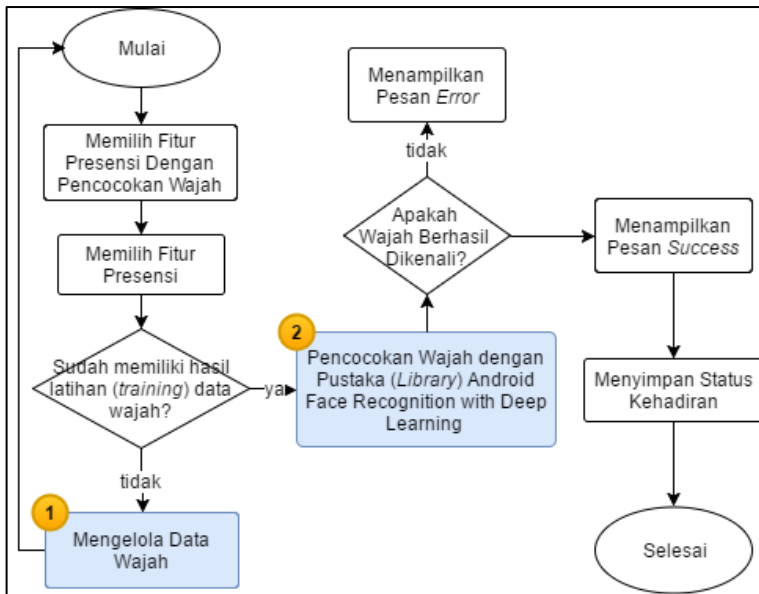
3.2.5 Perancangan Proses

Perancangan proses pada aplikasi Sikemas akan dijelaskan melalui diagram alir pada Gambar 3.16, Gambar 3.17. Pada subbab ini akan dijelaskan tahap proses yang dilakukan oleh aplikasi dengan lebih mendetil melalui tahapan-tahapan proses di setiap prosesnya.

3.2.5.1 Rancangan Alir Proses Utama Perangkat Lunak

Pada Gambar 3.16 dijelaskan alir proses perangkat lunak secara keseluruhan. Tahapan proses yang dilakukan antara lain:

1. Pada awal alir proses, mahasiswa memilih fitur presensi dengan menggunakan metode pencocokan wajah.
2. Mahasiswa memilih fitur presensi untuk memulai proses presensi kehadiran.



Gambar 3.16 Diagram Alir Proses Utama Perangkat Lunak

3. Jika aplikasi memiliki data hasil dari latihan (*training*) data wajah mahasiswa yang terlogin maka aplikasi memulai fitur presensi dan melakukan proses pencocokan wajah dengan pustaka (*library*) Android Face Recognition with Deep Learning.
4. Jika wajah mahasiswa tidak berhasil dikenal maka akan muncul pesan *error*.
5. Jika wajah mahasiswa berhasil dikenali dan dicocokkan maka aplikasi menyimpan status presensi kehadiran mahasiswa dan dikirim ke sistem server dan proses utama perangkat lunak selesai.

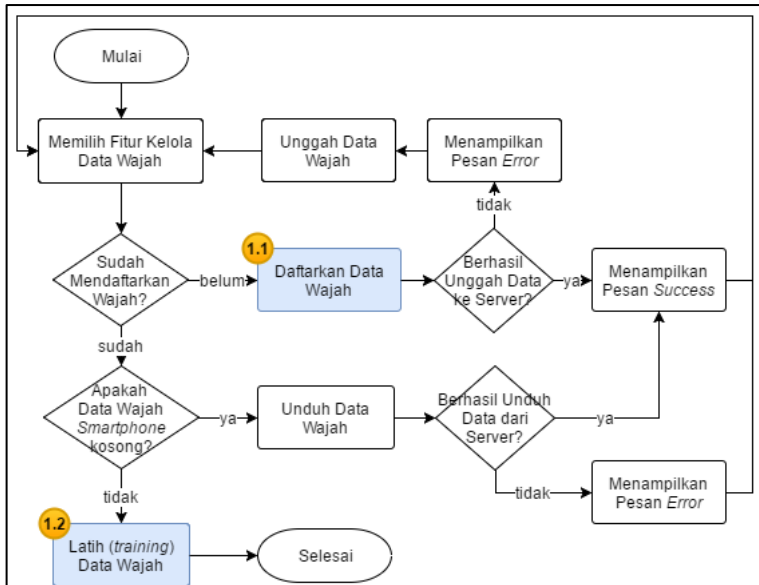
Pada diagram alir terdapat 2 proses yang dapat didekomposisi yaitu proses “Mengelola Data Wajah” yang ditunjukkan pada Gambar 3.16 nomor 1 dan proses pencocokan wajah dengan pustaka (*library*) Android Face Recognition with Deep Learning yang ditunjukkan pada Gambar 3.16 nomor 2.

3.2.5.2 Rancangan Alir Proses Mengelola Data Wajah

Gambar 3.17 merupakan gambar rancangan diagram alir untuk proses “Mengelola Data Wajah” dari dekomposisi proses utama perangkat lunak pada Gambar 3.16. Pada diagram ini memiliki 2 buah proses yang dapat didekomposisi proses-proses di dalamnya. Proses-proses tersebut antara lain proses “Daftarkan Data Wajah” yang ditunjukkan pada Gambar 3.17 nomor 1.1 dan proses “Latih (*training*) Data Wajah” yang ditunjukkan pada Gambar 3.17 nomor 1.2.

Tahapan proses dari diagram alir Gambar 3.17 antara lain:

1. Mahasiswa memilih fitur kelola data wajah untuk memulai proses mengelola data wajah.
2. Proses daftarkan data wajah dijalankan ketika mahasiswa belum pernah mendaftarkan wajah. Setelah proses ini dilakukan, tahap berikutnya aplikasi mengunggah data wajah secara otomatis.
3. Proses unduh data wajah dapat dilakukan ketika mahasiswa sudah pernah mendaftarkan wajah sebelumnya tetapi aplikasi tidak menemukan data wajah yang tersimpan dalam *smartphone* mahasiswa.
4. Jika proses unduh data wajah berhasil maka aplikasi akan menampilkan pesan success dan kembali ke fitur kelola data wajah.
5. Jika proses unduh data wajah gagal maka aplikasi akan menampilkan pesan error dan kembali ke fitur kelola data wajah.
6. Proses latihan (*training*) data wajah dapat dilakukan ketika mahasiswa telah melakukan pendaftaran sebelumnya dan aplikasi dapat menemukan data wajah yang tersimpan di *smartphone*.
7. Setelah proses latihan (*training*) data wajah berhasil, hal ini menandakan proses mengelola data wajah selesai.



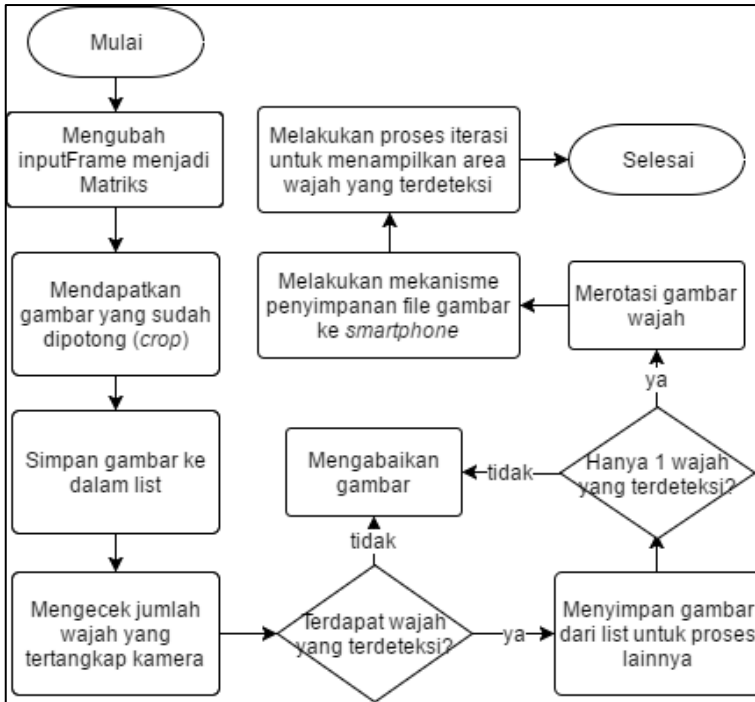
Gambar 3.17 Diagram Alir Mengelola Data Wajah

3.2.5.3 Rancangan Alir Proses Daftarkan Data Wajah

Pada bagian ini dirancang alir proses dari proses daftarkan data wajah pada aplikasi Sikemas. Proses ini digambarkan pada Gambar 3.18. Proses daftarkan data wajah merupakan hasil dekomposisi dari proses mengelola data wajah yang ditunjukkan pada Gambar 3.17 nomor 1.1.

Tahapan proses dari diagram alir Gambar 3.18 antara lain:

1. Kamera *smartphone* mendeteksi gambar wajah dalam bentuk *inputFrame* yang kemudian diubah menjadi matriks.
2. Aplikasi mendapatkan gambar wajah yang sudah dipotong dari fungsi *getCroppedImage* dari pustaka (*library*) yang dipakai.
3. Aplikasi menyimpan gambar ke dalam sebuah list matriks.
4. Aplikasi melakukan pengecekan jumlah wajah yang tertangkap kamera dalam list matriks pada tahap ke-3.



Gambar 3.18 Diagram Alir Daftarkan Data Wajah

5. Jika terdapat wajah yang terdeteksi maka aplikasi akan mengambil data wajah di dalam list untuk dipakai pada proses lainnya.
6. Jika tidak terdapat wajah yang terdeteksi maka aplikasi akan mengabaikan data wajah di dalam list.
7. Data wajah yang diambil pada tahap 5 kemudian dilakukan pengecekan lagi apakah jumlah wajah yang diambil hanya 1 wajah. Jika jumlah wajah yang diambil dalam 1 data wajah hanya 1 buah maka data wajah akan dirotasikan dalam sudut pengenalan wajah yang optimal. Jika jumlah wajah lebih 1 wajah maka data wajah yang sedang dicek akan diabaikan.

8. Setelah pengecekan data wajah, kemudian aplikasi akan melakukan mekanisme penyimpanan file gambar wajah ke penyimpanan internal *smartphone*.
9. Aplikasi menampilkan area wajah yang terdeteksi oleh kamera beserta perhitungan wajah yang berhasil diambil.
10. Setelah proses pengambilan wajah berhasil. Hal ini menandakan proses pendaftaran data wajah telah selesai.

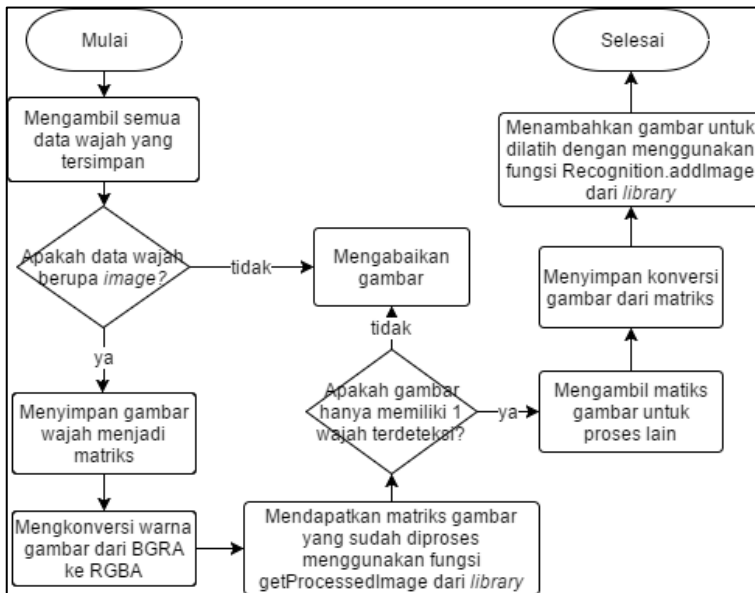
3.2.5.4 Rancangan Alir Proses Latih Data Wajah

Pada bagian ini dirancang alir proses dari proses latih data wajah pada aplikasi Sikemas. Proses ini digambarkan pada Gambar 3.19. Proses latih data wajah merupakan hasil dekomposisi dari proses mengelola data wajah yang ditunjukkan pada Gambar 3.17 nomor 1.2.

Tahapan proses dari diagram alir Gambar 3.19 antara lain:

1. Proses latih data wajah diawali dengan pengambilan semua data wajah yang tersimpan meliputi data wajah mahasiswa yang terlogin dan data wajah pembanding yang sudah disiapkan di dalam aplikasi.
2. Dilakukan pengecekan apakah data wajah yang diambil berupa *file gambar/image*.
3. Jika data wajah berupa *file gambar/image* maka aplikasi akan menyimpan data wajah ke dalam bentuk matriks.
4. Jika data wajah bukan berupa *file gambar/image* maka aplikasi akan mengabaikan gambar.
5. Gambar wajah yang tersimpan pada tahap nomor 3, kemudian dikonversikan warnanya dari BGRA ke RGBA.
6. Data wajah yang tersimpan kemudian diproses menggunakan fungsi `getProcessedImage` dari pustaka (*library*) yang digunakan, kemudian data wajah yang telah diproses diambil matriksnya.
7. Aplikasi melakukan pengecekan jumlah data wajah yang terdeteksi pada tahap 6.

8. Jika jumlah wajah yang terambil dalam 1 data wajah hanya 1 wajah yang terdeteksi maka matriks gambar akan diambil untuk proses lain.
9. Jika tidak atau lebih dari 1 wajah yang terdeteksi maka data wajah akan diabaikan.
10. Matriks gambar dari tahap 8 kemudian akan dikonversikan menjadi gambar.
11. Setelah gambar dikonversi kemudian aplikasi akan melakukan fungsi `Recognition.addImage` dari pustaka (*library*) untuk menambahkan gambar untuk dilatih.
12. Setelah gambar dilatih hingga selesai kemudian proses alir latih data wajah selesai.



Gambar 3.19 Diagram Alir Latih Data Wajah

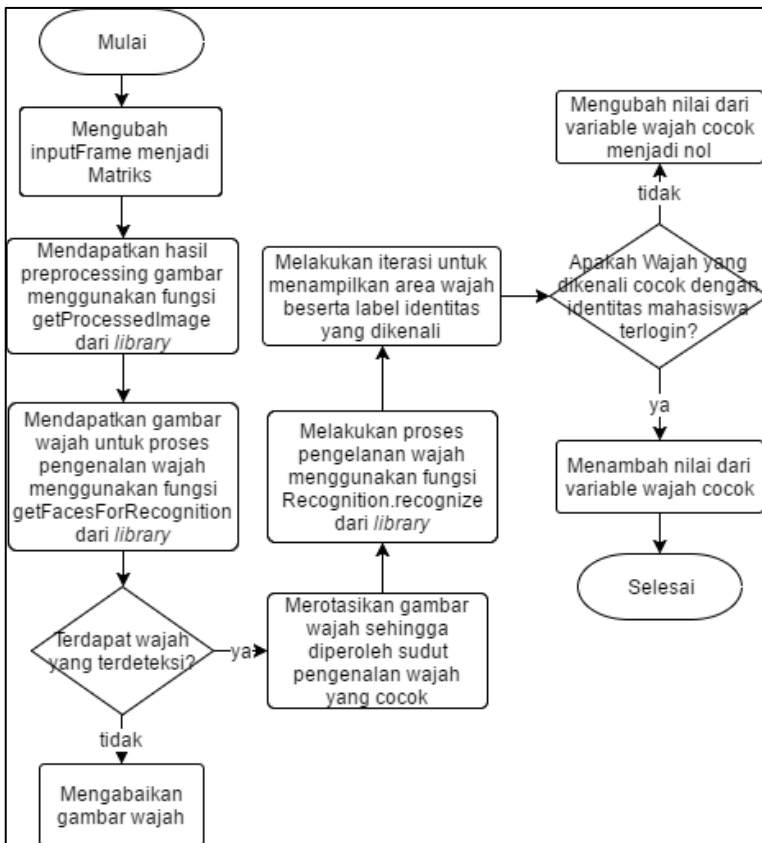
3.2.5.5 Rancangan Alir Proses Pencocokan Wajah Dengan Pustaka Android Face Recognition With Deep Learning

Pada bagian ini dirancang alir proses pencocokan wajah dengan pustaka Android Face Recognition with Deep Learning pada aplikasi Sikemas. Proses ini digambarkan pada Gambar 3.20. Proses pencocokan wajah ini merupakan hasil dekomposisi dari proses utama perangkat lunak yang ditunjukkan pada Gambar 3.16 nomor 2.

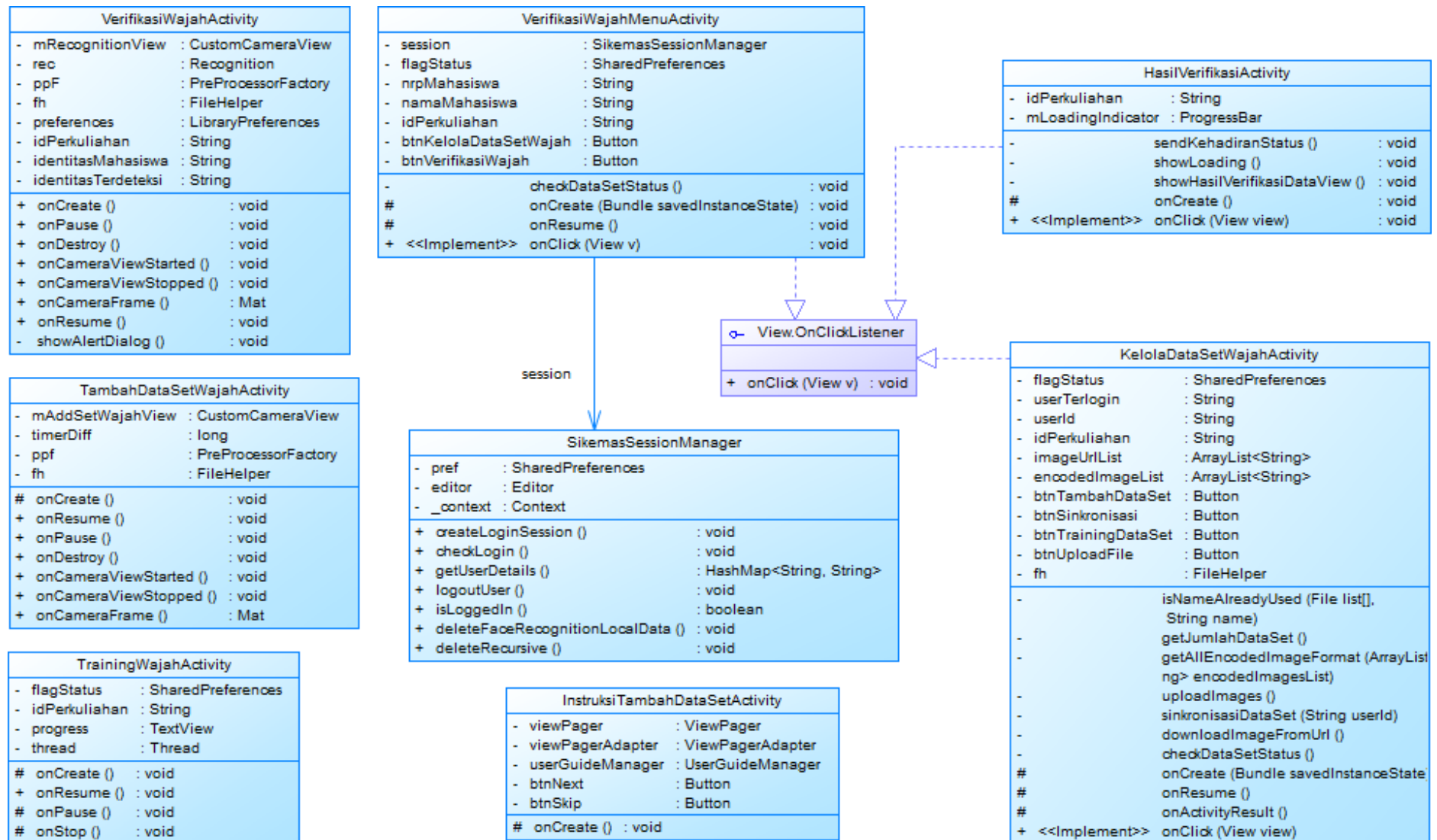
Tahapan proses dari diagram alir Gambar 3.20 antara lain:

1. Kamera *smartphone* mendeteksi gambar wajah dalam bentuk `inputFrame` yang kemudian diubah menjadi matriks.
2. Aplikasi mendapatkan hasil preprocessing gambar menggunakan fungsi `getProcessedImage` dari pustaka (*library*) yang digunakan.
3. Aplikasi mendapatkan gambar wajah untuk proses pengenalan wajah menggunakan fungsi `getFacesForRecognition` dari pustaka (*library*) yang digunakan.
4. Aplikasi melakukan pengecekan apakah terdapat wajah yang terdeteksi.
5. Jika ada wajah yang terdeteksi maka gambar wajah akan dirotasikan hingga memperoleh sudut pengenalan wajah yang cocok.
6. Jika tidak ada wajah yang terdeteksi maka gambar wajah akan diabaikan.
7. Dari tahap 5, kemudian aplikasi akan melakukan proses pengenalan wajah menggunakan fungsi `Recognition.recognize` dari pustaka (*library*) yang dipakai.
8. Aplikasi melakukan iterasi untuk menampilkan area wajah beserta label identitas yang dikenali.
9. Aplikasi melakukan pengecekan apakah wajah yang dikenali cocok dengan identitas mahasiswa yang terlogin dalam aplikasi.

10. Jika wajah yang dikenali cocok dengan identitas mahasiswa yang terlogin maka aplikasi akan menambah variable wajah cocok.
11. Jika wajah yang dikenali tidak cocok dengan identitas mahasiswa yang terlogin maka aplikasi akan mengubah variable wajah cocok menjadi nol.
12. Setelah proses pengenalan wajah dilakukan maka proses pencocokan wajah selesai.



Gambar 3.20 Diagram Alir Proses Pencocokan Wajah



Gambar 3.21 Diagram Kelas

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam aplikasi Sikemas dan disertai dengan kode sumber masing-masing fungsi utama.

4.1 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak ini dijelaskan pada Tabel 4.1.

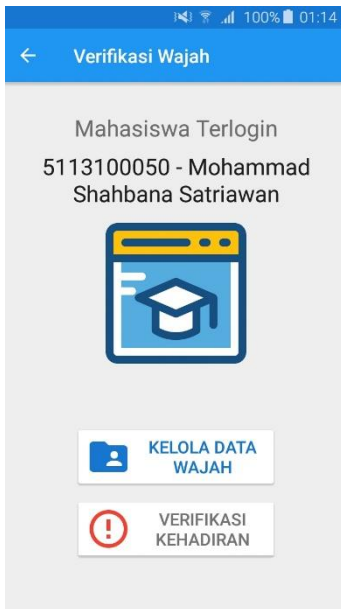
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat Keras	Komputer	Lenovo Z410
	Prosesor	Intel® Core™ i5-4200M CPU @2.50GHz (2.60GHz)
	Memori Primer	8 GB
	Memori Sekunder	1000 GB
Perangkat Lunak	Sistem Operasi	Windows 10 Education 64-bit
	Perangkat Lunak	Android Studio v2.3.1, Justinmind Prototyping tool, InvisionApp, Sybase PowerDesigner 16.5, Microsoft Word 2016

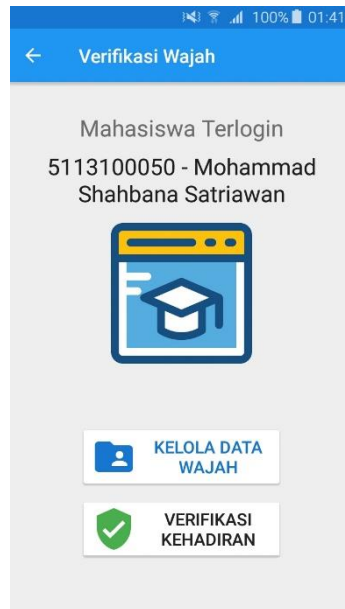
4.2 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna aplikasi mobile Android ini menggunakan jenis file XML yang dibangun dan diimplementasikan dalam lingkungan kerja Android Studio. Penjelasan implementasi antarmuka aplikasi ditunjukkan melalui tampilan halaman XML yang sudah dirender oleh Android Studio berikut sesuai dengan rancangan antarmuka pada bagian bab III tugas akhir ini.

4.2.1 Implementasi Halaman Antarmuka Menu Utama Verifikasi Wajah



Gambar 4.1 Implementasi Halaman Antarmuka Menu Utama Verifikasi Wajah (sebelum data wajah dilatih)



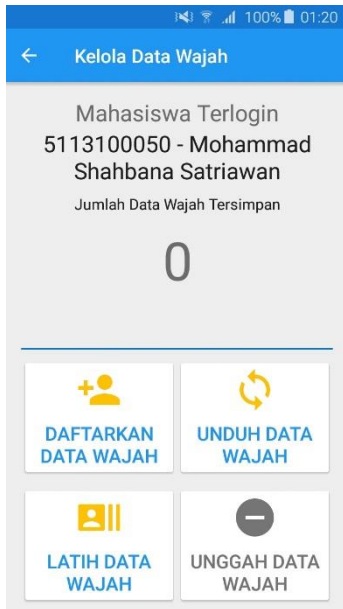
Gambar 4.2 Implementasi Halaman Antarmuka Menu Utama Verifikasi Wajah (setelah data wajah dilatih)

Halaman antarmuka menu utama verifikasi wajah pada Gambar 4.1 dan Gambar 4.2 merupakan halaman yang pertama kali diakses oleh mahasiswa ketika memilih pilihan verifikasi kehadiran menggunakan metode pencocokan wajah yang dibahas oleh sistem lain. Pada halaman ini terdapat informasi pengguna yang terlogin dalam aplikasi ini serta dua buah pilihan menu, pilihan kelola data wajah dan pilihan presensi. Di halaman ini terdapat informasi tambahan berupa gambar status dari pilihan presensi. Gambar status pilihan presensi pada Gambar 4.1 menunjukkan bahwa perangkat lunak tidak dapat menjalankan fitur presensi perangkat lunak belum memiliki data hasil latihan data wajah sebagai data acuan untuk proses pencocokan wajah. Gambar pilihan tombol verifikasi kehadiran pada Gambar 4.2 sebelah kanan menunjukkan bahwa perangkat lunak telah memiliki data hasil dari latihan data wajah dan telah siap untuk melakukan proses pencocokan wajah.

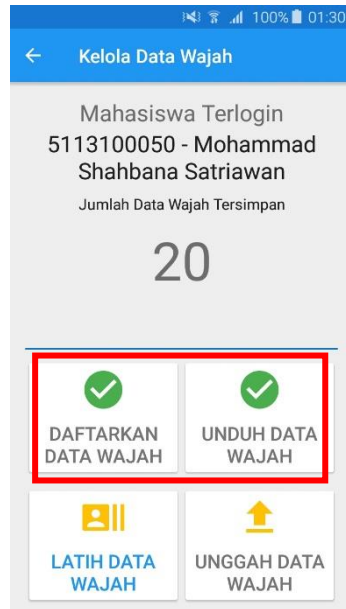
4.2.2 Implementasi Halaman Antarmuka Kelola Data Wajah

Halaman antarmuka kelola data wajah dapat dilihat pada Gambar 4.3, Gambar 4.4, Gambar 4.5, dan Gambar 4.6. Halaman ini memuat 4 pilihan fitur kelola data set wajah mahasiswa yang terlogin dalam aplikasi. Fitur yang dimiliki antara lain fitur tambah data set wajah, fitur unduh data set wajah, fitur training data set wajah, dan fitur unggah data set wajah. Pada halaman ini terdapat indikator jumlah data set wajah yang tersimpan pada internal penyimpanan *smartphone* yang digunakan serta status dari pilihan fitur yang disediakan. Pada Gambar 4.3 ditampilkan halaman antarmuka kelola data set wajah pada saat kondisi pertama kali aplikasi dijalankan. Kondisi ini ditampilkan ketika belum ada data set wajah yang tersimpan pada *smartphone*. Pada Gambar 4.4 ditampilkan perubahan gambar pada pilihan fitur daftar wajah dan unduh data wajah. Perubahan ini terjadi ketika pendaftaran data wajah telah dilakukan. Kondisi ini juga terjadi ketika data set wajah yang tersimpan pada internal penyimpanan *smartphone* memiliki jumlah yang sama dengan yang tersimpan di sistem server.

Perubahan gambar ini menandakan fitur pendaftaran data wajah dan unduh data wajah tidak perlu dipilih karena kondisi sudah terpenuhi



Gambar 4.3 Implementasi Halaman Antarmuka Kelola Data Wajah (1)



Gambar 4.4 Implementasi Halaman Antarmuka Kelola Data Wajah (2)

Tampilan pada Gambar 4.5 terjadi ketika jumlah data set wajah yang tersimpan di internal penyimpanan *smartphone* tidak sama jumlahnya dengan yang berada di sistem server. Hal ini dapat terjadi ketika terjadi kesalahan dalam penyimpanan data wajah ke internal penyimpanan *smartphone* yang menyebabkan jumlah data set wajah yang tersimpan dalam *smartphone* berbeda dengan yang tersimpan di sisi server. Tampilan pada Gambar 4.6 adalah tampilan dimana semua fitur kelola data set wajah telah lengkap.

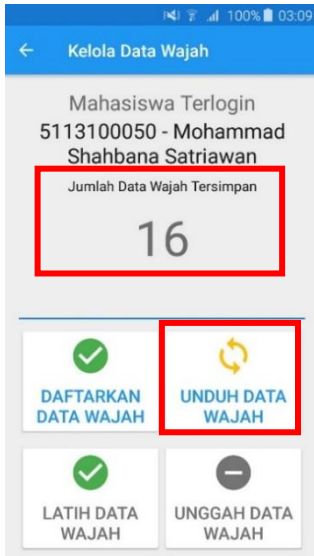
Fitur kelola data set wajah dianggap lengkap ketika mahasiswa telah mendaftarkan data wajah dan berhasil mengunggahnya ke server serta telah memilih fitur melatih data set wajah.

4.2.3 Implementasi Halaman Antarmuka Mendaftarkan Data Wajah

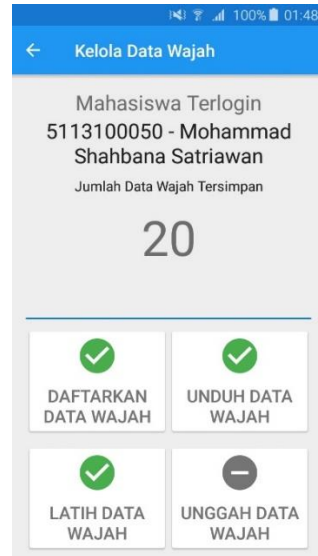
Halaman antarmuka mendaftarkan data set wajah ditampilkan pada Gambar 4.7 dan Gambar 4.8. Pada Gambar 4.7 ditunjukkan tampilan ketika tidak terdapat wajah yang terdeteksi sehingga tampilan antarmuka pada layar *smartphone* tidak menunjukkan adanya wajah yang terdeteksi dan penghitung wajah terambil akan selalu bernilai nol atau berhenti ketika tidak ada wajah yang terdeteksi. Pada Gambar 4.8 ditunjukkan tampilan antarmuka pada layar *smartphone* ketika terdapat wajah yang terambil. Pada tampilan ini terjadi proses pengambilan foto wajah mahasiswa yang terdeteksi lalu menyimpannya di penyimpanan internal *smartphone*. Pada aplikasi yang dibangun hanya dibutuhkan 20 buah foto yang diambil sehingga halaman antarmuka ini akan ditutup secara otomatis ketika data wajah yang terambil sudah mencapai 20 buah foto.

4.2.4 Implementasi Halaman Antarmuka Melatih Data Wajah

Halaman antarmuka melatih data set wajah ditampilkan pada Gambar 4.9. Pada tampilan ini akan ditunjukkan proses pelatihan data set wajah yang telah terambil dengan data set pembandingan yang telah disiapkan oleh aplikasi.



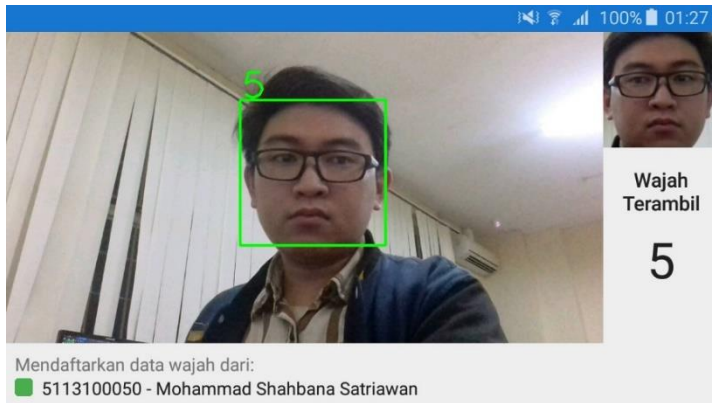
Gambar 4.5 Implementasi Halaman Antarmuka Kelola Data Set (3)



Gambar 4.6 Implementasi Halaman Antarmuka Kelola Data Set (4)



Gambar 4.7 Implementasi Halaman Antarmuka Tambah Data Set Wajah (Tidak Terdeteksi)



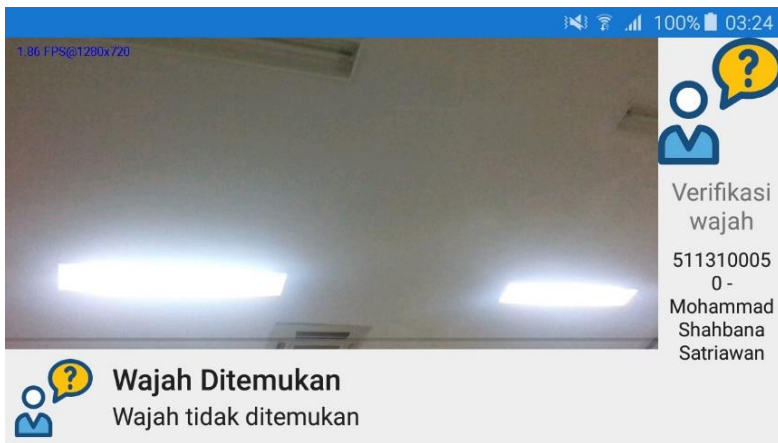
Gambar 4.8 Implementasi Halaman Antarmuka Tambah Data Set Wajah (Terdeteksi)



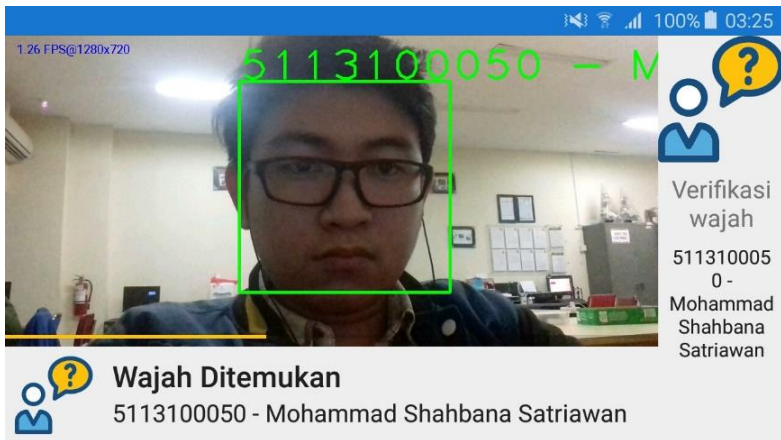
Gambar 4.9 Implementasi Halaman Antarmuka Training Data Set Wajah

4.2.5 Implementasi Halaman Antarmuka Presensi

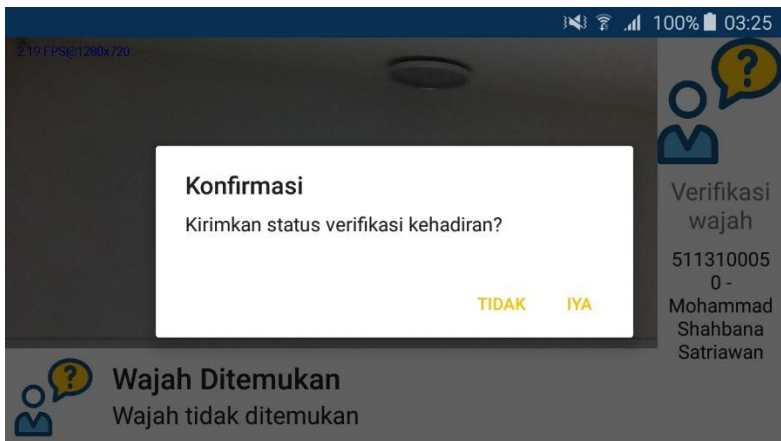
Implementasi dari halaman antarmuka presensi ditunjukkan pada Gambar 4.10, Gambar 4.11, dan Gambar 4.12. Tampilan pada Gambar 4.10 ditampilkan ketika tidak ada wajah yang terdeteksi oleh kamera *smartphone*. Pada Gambar 4.11 ditunjukkan tampilan ketika terdapat wajah yang terdeteksi dan dikenali oleh aplikasi. Pada gambar ini juga ditampilkan *progress bar* untuk menunjukkan sejauh mana proses pengenalan wajah dilakukan oleh aplikasi. Pada Gambar 4.12 ditampilkan sebuah *dialog* untuk konfirmasi pengiriman status kehadiran mahasiswa. Tampilan ini akan ditampilkan ketika *progress bar* dalam proses pengenalan wajah pada tampilan Gambar 4.11 telah penuh/selesai.



**Gambar 4.10 Implementasi Halaman Antarmuka Presensi
(Wajah Tidak Ditemukan)**



Gambar 4.11 Implementasi Halaman Antarmuka Presensi (Wajah Terdeteksi dan Cocok)



Gambar 4.12 Implementasi Halaman Antarmuka Presensi (Konfirmasi Kehadiran)

4.3 Implementasi Kasus Penggunaan dan Alir Proses

Implementasi kasus penggunaan dan alir proses aplikasi ini menggunakan bahasa pemrograman Java yang dibangun dengan menggunakan Android Studio. Pada subbab ini akan dijelaskan dan ditampilkan kode sumber sesuai dengan kasus penggunaan pada bab III.

4.4 Implementasi Kasus Penggunaan Mendaftarkan Data Wajah

Implementasi kasus penggunaan mendaftarkan data wajah akan dibagi menjadi 2 tahap pendaftaran. Tahap pertama adalah tahap dimana aplikasi mengambil wajah yang terdeteksi oleh kamera. Tahap kedua adalah tahap dimana aplikasi mengirimkan data wajah yang terambil ke sistem server. Implementasi dari kasus penggunaan ini dapat dilihat pada Kode Sumber 4.1 dan Kode Sumber 4.2.

```

1. public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame
   inputFrame) {
2.     Mat imgRgba = inputFrame.rgba();
3.     Mat imgCopy = new Mat();
4.     imgRgba.copyTo(imgCopy);
5.     // Selfie / Mirror mode
6.     if (front_camera) {
7.         Core.flip(imgRgba, imgRgba, 1);
8.     }
9.     long time = new Date().getTime();
10.    if ((method == MANUALLY) || (method == TIME) && (lastTime
        + timerDiff < time)) {
11.        lastTime = time;
12.        final List<Mat> images = ppF.getCroppedImage(imgCopy);
13.        if (images != null && images.size() == 1) {
14.            Mat img = images.get(0);
15.            if (img != null) {
16.                Rect[] faces = ppF.getFacesForRecognition();
17.                if ((faces != null) && (faces.length == 1)) {

```



```

18.         faces = MatOperation.rotateFaces(imgRgba, fac
es, ppF.getAngleForRecognition());
19.         if (((method == MANUALLY) && capturePressed)
|| (method == TIME)) {
20.             final MatName m = new MatName(name + "_" +
total, img);
21.             if (folder.equals("Test")) {
22.                 String wholeFolderPath = fh.TEST_PATH + na
me + "/" + subfolder;
23.                 new File(wholeFolderPath).mkdirs();
24.                 fh.saveMatToImage(m, wholeFolderPath + "/"
);
25.             } else {
26.                 String wholeFolderPath = fh.TRAINING_PATH
+ name;
27.                 new File(privatePath, wholeFolderPath).mkd
irs();
28.                 fh.saveMatToImage(m, wholeFolderPath + "/"
);
29.             }
30.             for (int i = 0; i < faces.length; i++) {
31.                 MatOperation.drawRectangleAndLabelOnPre
view(imgRgba, faces[i], String.valueOf(total + 1), front_cam
era);
32.                 runOnUiThread(new Runnable() {
33.                     @Override
34.                     public void run() {
35.                         Mat mat = m.getMat();
36.                         Bitmap bitmap = Bitmap.createBitmap(m
at.cols(), mat.rows(), Bitmap.Config.ARGB_8888);
37.                         Utils.matToBitmap(mat, bitmap);
38.                         capturedImagePreview.setImageBitmap(b
itmap);
39.                     }
40.                 });
41.             }
42.             total++;
43.             runOnUiThread(new Runnable() {
44.                 @Override
45.                 public void run() {
46.                     counter.setText(String.valueOf(total)
);
47.                 }

```

```

48.         });
49.         if (total == numberOfPictures) {
50.             Intent returnIntent = new Intent(TambahD
                ataSetWajahActivity.this, KelolaDataSetWajahActivity.class);
51.             returnIntent.putExtra("result_message",
                "Berhasil menambahkan data set wajah");
52.             returnIntent.putExtra("number_of_picture
                s", total);
53.             setResult(Activity.RESULT_OK, returnInte
                nt);
54.             flagStatus.edit().putBoolean("upload_fla
                g", false).apply();
55.             flagStatus.edit().putBoolean("training_f
                lag", false).apply();
56.             finish();
57.         }
58.         capturePressed = false;
59.     } else {
60.         for (int i = 0; i < faces.length; i++) {
61.             MatOperation.drawRectangleOnPreview(imgRgba, f
                aces[i], front_camera);
62.         }
63.     }
64. }
65. }
66. }
67. }
68. return imgRgba;
69. }

```

Kode Sumber 4.1 Mengambil wajah yang terdeteksi oleh kamera

Tabel 4.2 Penjelasan Kode Sumber 4.1

No. Baris	Kegunaan
1	Fungsi untuk mendeteksi wajah yang tertangkap pada frame kamera
2	Menampung rgba dari inputFrame kedalam bentuk matriks imgRgba

No. Baris	Kegunaan
3	Menginstansiasi Matriks baru bernama img
4	Menyalin matriks imgRgba ke matriks img
5 - 8	Melakukan proses mencerminkan tampilan kameran depan dari <i>smartphone</i> .
12	Menampung hasil crop gambar ke dalam list bernama images
13	Mengecek jumlah wajah yang tertangkap oleh kamera
14 - 16	Mengambil wajah pada indeks ke 0 dari list image kemudian melakukan cek adanya wajah atau tidak, jika iya maka gambar akan terambil
17	Mengecek jika hanya terdapat 1 wajah saja yang terdeteksi oleh kamera
18	Melakukan rotasi terhadap gambar yang terambil.
20 - 29	Melakukan mekanisme penyimpanan file gambar yang terambil ke internal penyimpanan <i>smartphone</i> .
30 - 41	Melakukan proses iterasi untuk penggambaran area wajah yang terdeteksi disertai dengan label jumlah data yang terambil pada setiap wajah yang terdeteksi
42 - 48	Mengubah tampilan jumlah wajah yang terambil.
49 - 58	Melakukan perubahan - perubahan yang dilakukan ketika jumlah wajah yang terambil telah mencapai 20 wajah.
59 - 69	Melakukan iterasi untuk penggambaran ulang area wajah yang terdeteksi

```

1. private void uploadImages(final ArrayList<String> encodedImagesList) {
2.     StringRequest stringRequest;
3.     RequestQueue requestQueue = Volley.newRequestQueue(this);
4.     //Showing the progress dialog
5.     progressDialog.setMessage("Mengunggah Data Set ke Server ... ");
6.     progressDialog.show();
7.
8.     final int[] numberOfPhotos = {0};
9.     for (int i = 0; i < encodedImagesList.size(); i++) {
10.         final int index = i;

```

```

11.     stringRequest = new StringRequest(Request.Method.POST,
    NetworkUtils.UPLOAD_DATASET_WAJAH_SIKEMAS,
12.     new Response.Listener<String>() {
13.         @Override
14.         public void onResponse(String response) {
15.             //Dismissing the progress dialog
16.             progressDialog.dismiss();
17.             numberOfPhotos[0]++;
18.             //Showing toast message of the response
19.             if (numberOfPhotos[0] == encodedImagesList.size()
    ) {
20.                 Toast.makeText(KelolaDataSetWajahActivity.this
    , "Berhasil Kirim Data Set ke Server",
21.                 Toast.LENGTH_SHORT).show();
22.                 flagStatus.edit().putBoolean("upload_flag", tr
    ue).apply();
23.             }
24.         }
25.     },
26.     new Response.ErrorListener() {
27.         @Override
28.         public void onErrorResponse(VolleyError volleyError)
    {
29.             // Dismissing the progress dialog
30.             progressDialog.dismiss();
31.             //Showing toast
32.             if (index == encodedImagesList.size())
33.                 Toast.makeText(KelolaDataSetWajahActivity.this,
    volleyError.getMessage(), Toast.LENGTH_LONG).show();
34.             flagStatus.edit().putBoolean("upload_flag", fal
    se).apply();
35.         }
36.     }) {
37.         @Override
38.         protected Map<String, String> getParams() throws
    AuthFailureError {
39.             // Get encoded Image
40.             String image = encodedImagesList.get(index);
41.             Map<String, String> params = new HashMap<>();
42.             params.put("image", image);
43.             params.put("image_name", userTerlogin + "_" +
    index);
44.             params.put("user_id", userId);

```

```

45.         return params;
46.     }
47. };
48.     VolleySingleton.getInstance(getApplicationContext())
        .addToRequestQueue(stringRequest);
49. }
50. }

```

Kode Sumber 4.2 Mengunggah data wajah ke sistem server

Tabel 4.3 Penjelasan Kode Sumber 4.2

No. Baris	Kegunaan
9	Melakukan iterasi sebanyak ukuran Arraylist image yang tersimpan di internal penyimpanan <i>smartphone</i>
11	Membuat request String untuk mengunggah data wajah ke sistem server
12 - 25	Menerima response ketika request string yang dikirimkan sukses / mendapatkan response.
26 - 36	Menerima response kesalahan ketika terjadi kesalahan pada request string yang dikirimkan
37 - 47	Mengirimkan parameter unggah data wajah ke server. Parameter yang dikirimkan antara lain encoded base 64 image, image_name, dan user_id dari mahasiswa yang melakukan pendaftaran data wajah
48	Menambahkan inputstream request ke request queue dari Volley.

4.4.1 Implementasi Kasus Penggunaan Melakukan Presensi

Implementasi kasus penggunaan memverifikasi kehadiran ditunjukkan pada Kode Sumber 4.3. Pada bagian verifikasi kehadiran ini, proses pengenalan wajah yang menjadi fokus utama dari penggunaan *library* yang dipakai dalam tugas akhir ini. Untuk proses verifikasi hanya menggunakan perhitungan pengambilan wajah oleh kamera sebanyak 10 kali secara berurutan.

```

1. public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {
2.     Mat imgRgba = inputFrame.rgba();
3.     Mat img = new Mat();
4.     imgRgba.copyTo(img);
5.     List<Mat> images = ppF.getProcessedImage(img, PreProcessorFactory.PreprocessingMode.RECOGNITION);
6.     Rect[] faces = ppF.getFacesForRecognition();
7.     // Selfie / Mirror mode
8.     if (front_camera) {
9.         Core.flip(imgRgba, imgRgba, 1);
10.    }
11.    if (images == null || images.size() == 0 || faces == null || faces.length == 0 || !(images.size() == faces.length)) {
12.        // skip
13.        runOnUiThread(new Runnable() {
14.            @Override
15.            public void run() {
16.                tvUserDetected.setText("Wajah tidak ditemukan");
17.                progress = 0;
18.                pbVerificationLoading.setProgress(progress);
19.            }
20.        });
21.        return imgRgba;
22.    } else {
23.        faces = MatOperation.rotateFaces(imgRgba, faces, ppF.getAngleForRecognition());
24.        for (int i = 0; i < faces.length; i++) {
25.            MatOperation.drawRectangleAndLabelOnPreview(imgRgba, faces[i], rec.recognize(images.get(i), ""), front_camera);
26.            identitasTerdeteksi = rec.recognize(images.get(i), "");
27.            if (identitasTerdeteksi.equals(identitasMahasiswa))
28.                progress += 10;
29.            else
30.                progress = 0;
31.        }
32.        runOnUiThread(new Runnable() {
33.            @Override
34.            public void run() {

```

```

35.         if (identitasTerdeteksi.equals(identitasMahasiswa)
36.     ) {
37.         tvUserDetected.setText(identitasTerdeteksi);
38.         pbVerificationLoading.setProgress(progress);
39.         if (progress == 100) {
40.             showAlertDialog(idPerkuliahan);
41.         }
42.     } else {
43.         progress = 0;
44.         pbVerificationLoading.setProgress(progress);
45.         tvUserDetected.setText("Tidak Cocok");
46.     }
47.     );
48.     return imgRgba;
49. }
50. }

```

Kode Sumber 4.3 Memproses pengenalan wajah

Tabel 4.4 Penjelasan Kode Sumber 4.3

No. Baris	Kegunaan
1	Fungsi untuk mengenali wajah yang tertangkap pada frame kamera
2	Menampung rgba dari inputFrame kedalam bentuk matriks imgRgba
3	Menginstansiasi Matriks baru bernama img
4	Menyalin matriks imgRgba ke matriks img
5	Melakukan proses pre processing terhadap matriks gambar yang terambil dari inputframe.
6	Mengambil area wajah untuk dikenali dalam bentuk kotak
7 - 10	Mencerminkan tampilan wajah yang terambil ketika menggunakan kamera depan <i>smartphone</i> .
11 - 22	Melakukan perubahan tampilan ketika tidak ada wajah yang terdeteksi.

No. Baris	Kegunaan
23	Merotasikan wajah sehingga diperoleh sudut pengenalan wajah yang cocok.
24	Menggambar area persegi beserta label wajah untuk wajah yang terdeteksi.
27 - 28	Menambah progress perhitungan verifikasi ketika wajah yang ditemukan oleh kamera sama dengan wajah mahasiswa yang terlogin.
29 – 31	Membuat progress perhitungan menjadi bernilai nol ketika tiap kali wajah gagal dikenali.
32 - 50	Mengatur tampilan aplikasi ketika wajah yang terdeteksi tidak dikenali.

4.4.2 Implementasi Alir Proses Latih (*training*) Data Wajah

Implementasi kasus penggunaan melatih data set wajah ditunjukkan dengan pada Kode Sumber 4.4. Proses melatih data set wajah akan dijalankan pada *thread* yang berbeda dengan main *thread* aplikasi. Proses latihan data ini dilakukan dengan menggunakan algoritma support vector machine (SVM) dari pustaka yang digunakan sehingga kode program yang akan ditunjukkan hanya berupa penggunaan algoritma dalam pustakanya saja. Data yang dilatih adalah data dari wajah mahasiswa yang tersimpan dalam internal penyimpanan *smartphone* dan juga data pembandingan yang sudah disiapkan oleh sistem yang dimasukkan ke dalam APK.

```

1. final Handler handler = new Handler(Looper.getMainLooper());
2. thread = new Thread(new Runnable() {
3.     public void run() {
4.         if (!Thread.currentThread().isInterrupted()) {
5.             PreProcessorFactory ppF = new PreProcessorFactory(getApplication
6.                 Context());
7.             PreferencesHelper preferencesHelper = new Preference
8.                 sHelper(getApplicationContext());
9.             String algorithm = preferencesHelper.getClassificati
10.                 onMethod();
11.             try {

```



```

9.         int counter = 1;
10.        FileHelper fileHelper = new FileHelper();
11.        fileHelper.createDataFolderIfNotExsiting();
12.        final File[] persons = fileHelper.getTrainingList
13.        ();
14.        AssetManager assetManager = getAssets();
15.        final String[] assetFiles = assetManager.list("Da
16.        taSetWajah");
17.        if (persons.length > 0 && assetFiles.length > 0)
18.        {
19.            Recognition rec = RecognitionFactory.getRecogniti
20.            onAlgorithm(getApplicationContext(),
21.            Recognition.TRAINING, algorithm);
22.            for (int i = 0; i < assetFiles.length; i++) {
23.                InputStream inputStream;
24.                OutputStream outputStream;
25.                inputStream = assetManager.open("DataSetWajah/
26.                " + assetFiles[i]);
27.                File outputFile = new File(FileHelper.TRAINING
28.                _PATH + "/" + "processedDataSet.png");
29.                outputStream = new FileOutputStream(outputFile
30.                );
31.                int read = 0;
32.                byte[] bytes = new byte[1024];
33.                while ((read = inputStream.read(bytes)) != -
34.                1) {
35.                    outputStream.write(bytes, 0, read);
36.                }
37.                outputStream.close();

```

Kode Sumber 4.4 Melatih data wajah mahasiswa dengan data wajah pembanding di penyimpanan internal smartphone (1)

Tabel 4.5 Penjelasan Kode Sumber 4.4

No. Baris	Kegunaan
1	Menginstansiasikelas handler untuk menjembatani komunikasi antara <i>workerthread</i> dengan <i>main thread</i> .

No. Baris	Kegunaan
2	Menginstansiasi kelas thread untuk melakukan proses melatih data set wajah
5	Menginstansiasi kelas PreProcessorFactory (kelas dari <i>library</i>) untuk activity saat ini. Kelas PreProcessorFactory berfungsi sebagai kelas pembantu untuk pengambilan data-data yang akan diolah sebelum dan sesudah data diproses.
6	Menginstansiasi kelas PreferencesHelper (kelas dari <i>library</i>) untuk activity saat ini. Kelas PreferencesHelper berfungsi sebagai kelas pembantu untuk mendapatkan pengaturan-pengaturan dari metode yang akan dipakai.
7	Membuat variable algorithm bertipe data String untuk menampung nama metode klasifikasi yang dipakai.
10	Menginstansiasi kelas FileHelper (kelas dari <i>library</i>) untuk membantu dalam memanipulasi pembuatan file dan folder yang diperlukan dalam penggunaan <i>library</i> .
12	Membuat variable persons bertipe data Array of File untuk menampung semua data wajah mahasiswa yang disimpan di internal penyimpanan <i>smartphone</i> (file hasil dari pendaftaran/unduhan data wajah).
13 – 14	Mengambil semua lokasi dari data set pembandingan yang telah disediakan di dalam aplikasi.
15	Percabangan untuk memastikan terdapat data wajah mahasiswa dan data pembandingan yang tersimpan
16	Implementasi dari kelas Interface Recognition untuk mendapatkan algoritma recognition untuk training dari algoritma klasifikasi yang sudah ditentukan sebelumnya
18 - 29	Mendapatkan seluruh file data wajah pembandingan dari list file lokasi yang didapat dari baris kode 13 – 14.

```

30.         if (FileHelper.isFileAnImage(outputFile)) {
31.             Mat imgRgb = imread(outputFile.getAbsolutePath()
ath());
32.             Imgproc.cvtColor(imgRgb, imgRgb, Imgproc.COLOR_BGRA2RGBA);

```

```

33.         Mat processedDataSetImage = new Mat();
34.         imgRgb.copyTo(processedDataSetImage);
35.         List<Mat> images = ppF.getProcessedImage(processedDataSetImage, PreProcessorFactory.PreprocessingMode.RECOGNITION);
36.         if (images == null || images.size() > 1) {
37.             continue;
38.         } else {
39.             processedDataSetImage = images.get(0);
40.         }
41.         if (processedDataSetImage.empty()) {
42.             continue;
43.         }
44.         MatName m = new MatName("processedImage", processedDataSetImage);
45.         fileHelper.saveMatToImage(m, FileHelper.DAT
A_PATH);
46.         rec.addImage(processedDataSetImage, outputFile.getParent(), false);
47.         // Update screen to show the progress
48.         final int counterPost = counter;
49.         final int filesLength = assetFiles.length;
50.         progress.post(new Runnable() {
51.             @Override
52.             public void run() {
53.                 progress.append("Image " + counterPost
+ " of " + filesLength + " from " + "asset files" + " imported.\n");
54.             }
55.         });
56.         counter++;
57.     }

```

Kode Sumber 4.5 Melatih data wajah mahasiswa dengan data wajah pembandingan di penyimpanan internal smartphone (2)

Tabel 4.6 Penjelasan Kode Sumber 4.5

No. Baris	Kegunaan
30	Percabangan untuk nilai benar dari kondisi mengecek apakah file data wajah pembanding yang didapatkan berupa file gambar.
31	Membaca file gambar wajah yang didapat ke dalam variable <code>imgRgb</code> bertipe data <code>Mat</code> (kelas dari <i>library</i> <code>OpenCV</code>).
32	Mengkonversi warna gambar dari <code>BGRA</code> ke <code>RGBA</code> .
33	Menginstansiasikan kelas <code>Mat</code> baru bernama <code>processedDataSetImage</code>
34	Menyalin hasil konversi gambar pada baris program 32 ke <code>processedDataSetImage</code>
35	Menampung hasil dari pemrosesan gambar (pemrosesan gambar di dalam <i>library</i>) dari Matriks <code>processedDataSetImage</code> ke dalam sebuah list bernama <code>images</code> .
36 - 39	Mengabaikan file gambar data set yang memiliki lebih dari 1 gambar wajah yang terdeteksi atau tidak ada sama sekali. Program melakukan pengecekan ukuran list <code>image</code> lebih dari 1 atau kosong.
40 - 42	Mengabaikan <code>processedDataSetImage</code> yang kosong.
43	Memberi nama Matriks <code>processedDataSetImage</code> menjadi “ <code>processedImage</code> ”
44	Menyimpan matriks pada baris ke 43 menjadi gambar, dan menaruhnya sesuai lokasinya
45	Memproses data gambar (proses ekstraksi fitur di dalam <i>library</i> dengan metode yg sudah ditentukan)
46 - 57	Menampilkan proses training ke layar <i>smartphone</i>

```

58.         for (File person : persons) {
59.             if (person.isDirectory()) {
60.                 File[] files = person.listFiles();
61.                 for (File file : files) {
62.                     if (FileHelper.isFileAnImage(file)) {
63.                         Mat imgRgb = Imgcodecs.imread(file.get
AbsolutePath());

```

```

64.         Imgproc.cvtColor(imgRgb, imgRgb, Imgproc.COLOR_BGRA2RGBA);
65.         Mat processedImage = new Mat();
66.         imgRgb.copyTo(processedImage);
67.         List<Mat> images = ppF.getProcessedImages(processedImage, PreProcessorFactory.PreprocessingMode.RECOGNITION);
68.         if (images == null || images.size() > 1) {
69.             // More than 1 face detected cannot use this file for training
70.             continue;
71.         } else {
72.             processedImage = images.get(0);
73.         }
74.         if (processedImage.empty()) {
75.             continue;
76.         }
77.         // The last token is the name
78.         Folder name = PersonNameHelper.getFileFolder(file);
79.         String[] tokens = file.getParent().split("/");
80.         final String name = tokens[tokens.length - 1];
81.         MatName m = new MatName("processedImage", processedImage);
82.         fileHelper.saveMatToImage(m, FileHelper.DATA_PATH);
83.         rec.addImage(processedImage, name, false);
84.         // Update screen to show the progress
85.         final int counterPost = counter;
86.         final int filesLength = files.length;
87.         progress.post(new Runnable() {
88.             @Override
89.             public void run() {
90.                 progress.append("Image " + counterPost + " of " + filesLength + " from " + name + " imported.\n");
91.             }
92.         });

```

```

91.                                     });
92.                                     counter++;
93.                                 }
94.                            }
95.                       }
96.                   }

```

Kode Sumber 4.6 Melatih data wajah mahasiswa dengan data wajah pembanding di penyimpanan internal smartphone (3)

Tabel 4.7 Penjelasan Kode Sumber 4.6

No. Baris	Kegunaan
58 - 60	Mendapatkan seluruh file data wajah mahasiswa dari list file di dalam folder penyimpanan internal <i>smartphone</i> mahasiswa yang terlogin.
61	Iterasi untuk setiap file yang ada di dalam array files.
62	Percabangan untuk nilai benar dari kondisi mengecek apakah file data wajah mahasiswa yang didapatkan berupa file gambar.
63	Membaca file gambar wajah yang didapat ke dalam variable <i>imgRgb</i> bertipe data <i>Mat</i> (kelas dari <i>library</i> <i>OpenCV</i>).
64	Mengkonversi warna gambar dari <i>BGRA</i> ke <i>RGBA</i> .
65	Menginstansiasi kelas <i>Mat</i> baru bernama <i>processedImage</i>
66	Menyalin hasil konversi gambar pada baris program 65 ke <i>processedImage</i>
67	Menampung hasil dari pemrosesan gambar (pemrosesan gambar di dalam <i>library</i>) dari Matriks <i>processedImage</i> ke dalam sebuah list bernama <i>images</i> .
68 - 73	Mengabaikan file gambar data set yang memiliki lebih dari 1 gambar wajah yang terdeteksi atau tidak ada sama sekali. Program melakukan pengecekan ukuran list image lebih dari 1 atau kosong.
74 - 76	Mengabaikan <i>processedImage</i> yang kosong.
80	Memberi nama Matriks <i>processedImage</i> menjadi “ <i>processedImage</i> ”

No. Baris	Kegunaan
81	Menyimpan matriks pada baris ke 43 menjadi gambar, dan menaruhnya sesuai lokasi yang sudah ditentukan
82	Memproses data gambar (proses ekstraksi fitur di dalam <i>library</i> dengan metode yang sudah ditentukan)
85 - 96	Menampilkan proses training ke layar <i>smartphone</i>

```

97.         final Intent intent = new Intent(getApplicationContext(), VerifikasiWajahMenuActivity.class);
98.         intent.putExtra("id_perkuliahan", idPerkuliahan);
99.         intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
100.        if (rec.train()) {
101.            intent.putExtra("training", "Training successful");
102.            flagStatus.edit().putBoolean("training_flag", true).apply();
103.            flagStatus.edit().putBoolean("upload_flag", true).apply();
104.        } else {
105.            intent.putExtra("training", "Training failed");
106.            flagStatus.edit().putBoolean("training_flag", false).apply();
107.        }
108.        handler.post(new Runnable() {
109.            @Override
110.            public void run() {
111.                startActivity(intent);
112.                finish();
113.            }
114.        });
115.    } else {
116.        Thread.currentThread().interrupt();
117.    }
118.    } catch (IOException e) {
119.        e.printStackTrace();
120.    }
121.    }
122.    }

```

```
123.});
124.thread.start();
```

Kode Sumber 4.7 Melatih data wajah mahasiswa dengan data wajah pembanding di penyimpanan internal smartphone (4)

Tabel 4.8 Penjelasan Kode Sumber 4.7

No. Baris	Kegunaan
97	Menginstansiasi Intent untuk perpindahan dari activity saat ini menuju ke VerifikasiWajahMenuActivity
98	Menambahkan informasi tambahan idPerkuliah melalui intentExtra.
99	Menandai intent sebagai FLAG_ACTIVITY_CLEAR_TOP yang artinya ketika intent dilakukan maka aktivitas saat ini akan ditimpa (digantikan) dengan aktivitas intent yang dituju.
100	Melakukan <i>training</i> data (menggunakan <i>library</i> dengan metode yang telah ditentukan)
100 - 104	Memberikan informasi tambahan ketika <i>training</i> data berhasil dilakukan.
105 - 107	Memberikan informasi tambahan ketika terjadi kegagalan saat <i>training</i> data
108 - 115	Menjalankan handler setelah <i>workerthread</i> selesai
111	Melakukan perpindahan intent melalui <code>startActivity</code>
112	Menutup activity saat ini
115 - 117	Menginterupsi thread ketika kondisi pada bari program ke-15 tidak terpenuhi
124	Menjalankan thread

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk tugas akhir ini. Pengujian dilakukan pada kasus penggunaan dari sistem perangkat lunak.

5.1 Lingkungan Pengujian

Pada proses pengujian fungsionalitas perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam uji fungsionalitas ini dipakai untuk semua kasus penggunaan. Spesifikasi lingkungan pengujian akan dijelaskan pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak

Spesifikasi	Deskripsi
Jenis Perangkat	<i>Smartphone</i>
Merek Perangkat	Samsung Galaxi J3 2016
Sistem Operasi	Android Lollipop v5.1.1
Memori Internal	4 GB
RAM	2 GB

5.2 Pengujian Fungsionalitas

Pengujian fungsionalitas ini adalah pengujian fungsi-fungsi yang berjalan pada aplikasi berdasarkan kasus penggunaan. Pada subbab ini akan dijelaskan skenario pengujian yang digunakan. Pengujian fungsionalitas ini dilakukan dengan metode kotak hitam (*black box*), yaitu sebuah metode pengujian yang menekankan pada hasil keluaran program/aplikasi. Pengujian fungsionalitas akan dijelaskan pada Tabel 5.2 dan Tabel 5.3.

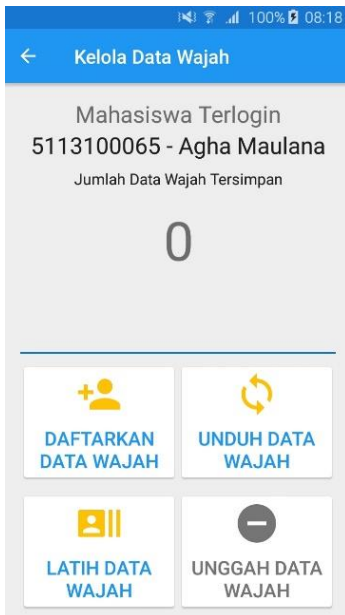
5.2.1 Pengujian Fungsionalitas Mendaftarkan Data Wajah

Pengujian fungsionalitas mendaftar data wajah melalui pengambilan foto wajah merupakan pengujian terhadap kemampuan aplikasi untuk mengambil data wajah pengguna yang terdeteksi, menyimpan ke penyimpanan internal *smartphone*, mengunggah data wajah ke sistem server, hingga menampilkan notifikasi keberhasilan pendaftaran data wajah. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.2.

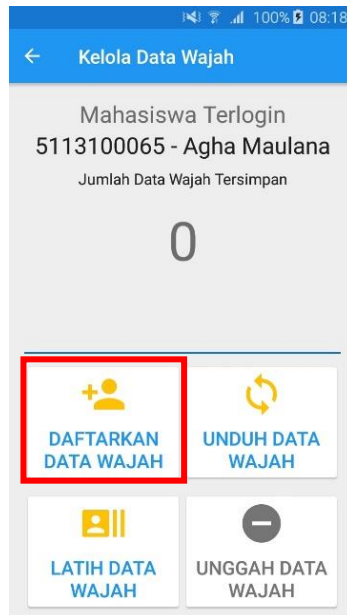
Tabel 5.2 Pengujian Fungsionalitas Mendaftarkan Data Wajah

ID	UJ.UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian mendaftar data wajah.
Tujuan Pengujian	Menguji pendaftaran data wajah menggunakan pengambilan foto wajah.
Skenario 1	<i>Pengguna mendaftar data wajah ketika awal menggunakan aplikasi dan tidak terjadi error unggah data.</i>
Kondisi Awal	Pengguna berada pada halaman kelola data wajah seperti pada Gambar 5.1. Aplikasi maupun sistem server belum memiliki data wajah mahasiswa yang terlogin dalam aplikasi.
Data Uji	Nama: Agha Maulana, NRP: 5113100065
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Daftarkan Data Wajah” seperti pada Gambar 5.2. 2. Pengguna memposisikan wajahnya hingga terdeteksi oleh aplikasi seperti pada Gambar 5.3. 3. Pengguna melakukan langkah nomor 2 sampai aplikasi mengambil data wajah sebanyak 20 kali.

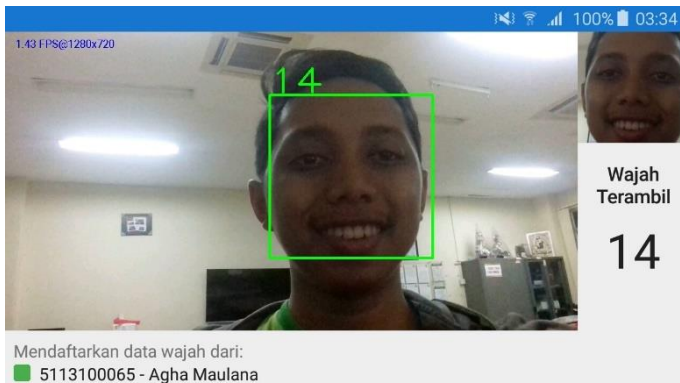
Hasil yang diharapkan	Aplikasi akan menampilkan notifikasi pendaftaran data wajah berhasil.
Hasil yang diperoleh	Aplikasi menampilkan notifikasi pendaftaran data wajah berhasil.
Hasil Pengujian	Berhasil
Kondisi Akhir	Data wajah berhasil disimpan di penyimpanan internal <i>smartphone</i> dan diunggah ke sistem server.
Skenario 2	<i>Pengguna sudah mendaftarkan data wajah tetapi terjadi error pada saat unggah data ke sistem server.</i>
Kondisi Awal	Pengguna berada pada halaman kelola data wajah dengan keadaan seperti pada Gambar 5.1 Aplikasi sudah memiliki data wajah mahasiswa yang terlogin Sistem server belum memiliki data wajah mahasiswa yang terlogin.
Data Uji	Nama: Agha Maulana, NRP: 5113100065
Langkah Pengujian	1. Pengguna menekan tombol “Unggah Data Wajah” seperti pada Gambar 5.2.
Hasil yang diharapkan	Aplikasi akan menampilkan notifikasi unggah data wajah berhasil.
Hasil yang diperoleh	Aplikasi menampilkan notifikasi unggah data wajah berhasil
Hasil Pengujian	Berhasil
Kondisi Akhir	Data wajah berhasil diunggah ke sistem server



Gambar 5.1 Tampilan Kelola Data Wajah Keadaan Awal



Gambar 5.2 Memilih Pilihan Daftarkan Data Wajah



Gambar 5.3 Memposisikan Wajah Hingga Terdeteksi oleh Aplikasi

5.2.2 Pengujian Fungsionalitas Melakukan Presensi

Pengujian fungsionalitas melakukan presensi merupakan pengujian terhadap kemampuan aplikasi untuk mengenali wajah pengguna yang terdeteksi, memverifikasi wajah, serta mengunggah status presensi mahasiswa ke sistem server hingga menampilkan halaman keberhasilan presensi. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pengujian Fungsionalitas Melakukan Presensi

ID	UJ.UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian melakukan presensi.
Tujuan Pengujian	Menguji presensi dengan menggunakan pengenalan foto wajah.
Skenario 1	<i>Pengguna melakukan presensi</i>
Kondisi Awal	Pengguna berada pada halaman menu utama verifikasi wajah seperti pada Gambar 5.4
Data Uji	Nama: Agha Maulana, NRP: 5113100065
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna menekan tombol “Verifikasi Kehadiran” seperti pada Gambar 5.5. 2. Pengguna memposisikan wajahnya hingga terdeteksi oleh aplikasi seperti pada Gambar 5.6. 3. Pengguna melakukan langkah 2 hingga 10 kali pengenalan wajah oleh aplikasi. 4. Pengguna memilih pilihan “iya” untuk mengkonfirmasi presensi.
Hasil yang diharapkan	Aplikasi akan menampilkan dialog untuk mengkonfirmasi presensi yang dilakukan. Aplikasi akan menampilkan halaman presensi berhasil dilakukan

Hasil yang diperoleh	Aplikasi menampilkan dialog untuk mengkonfirmasi presensi yang dilakukan seperti pada Gambar 5.8 Aplikasi menampilkan halaman presensi berhasil dilakukan.
Hasil Pengujian	Berhasil
Kondisi Akhir	Status presensi berhasil dikirimkan ke sistem server



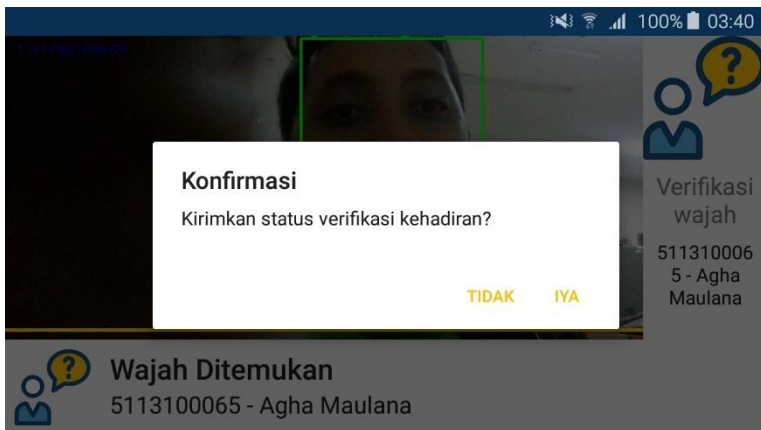
Gambar 5.4 Tampilan Verifikasi Wajah



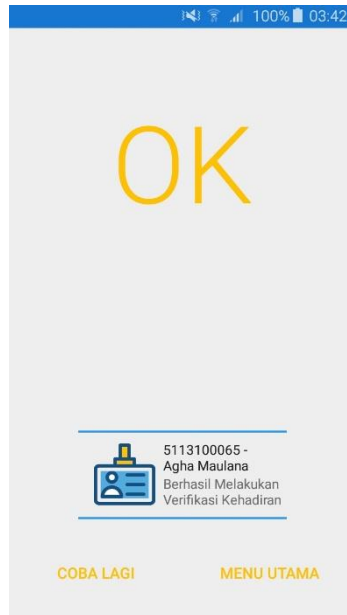
Gambar 5.5 Menekan Pilihan Verifikasi Kehadiran



Gambar 5.6 Tampilan Presensi dan Mendeteksi Wajah



Gambar 5.7 Tampilan Dialog Konfirmasi Presensi



Gambar 5.8 Tampilan Status Presensi

5.3 Pengujian Kegunaan

Pengujian kegunaan ini dilakukan untuk mengetahui tingkat kegunaan aplikasi terhadap pengguna. Kegunaan yang diuji meliputi kemudahan dalam melakukan verifikasi kehadiran dengan menggunakan pencocokan wajah.

Pengujian dilakukan dengan memberikan kesempatan pada pengguna untuk mencoba sendiri melakukan aplikasi yang telah dikembangkan dengan arahan dari pengembang aplikasi. Pengujian dilakukan dengan scenario terdapat 3 kelas dengan masing-masing kelas memiliki 5 mahasiswa sehingga terdapat 15 pengguna yang mencoba menggunakan aplikasi ini. Uji coba yang dilakukan adalah mahasiswa pada kelas tersebut melakukan verifikasi kehadiran mulai dari mendaftarkan data wajah hingga melakukan verifikasi kehadiran. Setelah pengujian selesai

pengguna diminta untuk mengisi kuesioner yang diberikan untuk mengetahui tanggapan pengguna. Rincian dari kuesioner dapat dilihat pada tabel Tabel 5.4.

Daftar pengguna yang melakukan pengujian dapat dilihat pada Tabel 5.5. Hasil dari kuesioner yang telah diisi oleh pengguna dikelompokkan berdasarkan masing-masing kelas. Penilaian kuesioner ini memiliki skala 1-6 dimana semakin kecil angka skala yang dipilih berarti semakin tidak setuju sedangkan semakin besar nilai angka skala yang dipilih berarti semakin sangat setuju dengan pernyataan kuesioner. Hasil kuesioner masing-masing kelas dapat dilihat pada Tabel 5.6, Tabel 5.7, dan Tabel 5.8. Rekapitulasi hasil kuesioner kelas A, B, dan C disajikan dalam bentuk persentase dari masing-masing pernyataan yang dapat dilihat pada tabel Tabel 5.9.

Tabel 5.4 Kuesioner Pengujian Kegunaan

No	Pernyataan	Penilaian					
		1	2	3	4	5	6
1	Menurut saya, aplikasi ini mudah untuk digunakan						
2	Menurut saya, presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (presensi di kertas)						
3	Menurut saya, lebih mudah dan efisien melakukan presensi dengan menggunakan aplikasi ini daripada sistem manual (presensi di kertas)						
4	Menurut saya, mudah menggunakan fitur presensi dengan pencocokan wajah						

No	Pernyataan	Penilaian					
		1	2	3	4	5	6
5	Menurut saya, mudah memasukkan data wajah ke dalam aplikasi						
6	Menurut saya, hasil presensi dengan fitur pencocokan wajah memiliki hasil yang akurat						
7	Menurut saya, presensi dengan menggunakan fitur pencocokan wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (presensi di kertas)						
8	Menurut saya, lebih mudah dan efisien melakukan presensi dengan menggunakan fitur pencocokan wajah dalam aplikasi ini daripada sistem manual (presensi di kertas)						

Keterangan:

1 = sangat tidak setuju

2 = tidak setuju

3 = cukup setuju

4 = setuju

5 = setuju sekali

6 = sangat setuju sekali

Tabel 5.5 Daftar Penguji Aplikasi

No	NRP	Nama Peserta	Smartphone dan Sistem Operasi	Kelas Skenario
1	5113100051	Mardiana Sekarsari	Sony C5 + Android Marshmallow	Kelas A

No	NRP	Nama Peserta	Smartphone dan Sistem Operasi	Kelas Skenario
2	5113100075	Achmad Rizky Haqiqi	Samsung Galaxy Note 3 + Android Marshmallow	Kelas A
3	5113100128	Relaci Aprilia Istiqomah	Sony Xperia Z5 + Android Nougat	Kelas A
4	5113100172	Afif Ishamsyah Hantriono	Samsung Galaxy A5 + Android Marshmallow	Kelas A
5	5115100062	Alvin Mudhoffar	Xiaomi Redmi Note 2 + Android Lollipop	Kelas A
6	5113100053	Lophita Y. Napitupulu	Samsung Galaxy J2 + Android Lollipop	Kelas B
7	5113100072	Ayu Kardina Sukmawati	Oppo A37 + Android Lollipop	Kelas B
8	5113100143	Nuning Septiana	Samsung Galaxy J2 + Android Lollipop	Kelas B
9	5113100173	Andi Akram Yusuf	Sony Xperia Z1 Compact + Android Lollipop	Kelas B
10	5113100174	Armirara Refa Aziz	Oppo A37 + Android Lollipop	Kelas B
11	5113100127	Luffi Aditya Sandy	Samsung Galaxy A3 + Android Marshmallow	Kelas C
12	5113100141	Laurensius Adi	Lenovo Vibe K5+ + Android Lollipop	Kelas C
13	5113100144	Andre Zachary Reinaldi	LG Google Nexus 5 + Android Marshmallow	Kelas C
14	5113100146	Kevin Zulkarnain Yuseti	Lenovo Vibe K5+ + Android Lollipop	Kelas C

No	NRP	Nama Peserta	Smartphone dan Sistem Operasi	Kelas Skenario
15	5113100015	Demsy Iman Mustasyar	Xiaomi 4 + Android Marshmallow	Kelas C

5.3.1 Pengujian Kegunaan Skenario Kelas A

Berikut ini adalah rincian pengujian kegunaan skenario kelas A:

- Nama Mata Kuliah : ERPL (Ekonomi Rekayasa Perangkat Lunak)
- Kelas : A
- Jam pelaksanaan : Pukul 15.00 – 17.30 WIB
- Ruang Kelas : IF-101

Tabel 5.6 adalah tabel hasil penilaian kuesioner skenario kelas A dari masing-masing mahasiswa pada setiap nomor pernyataan kuesioner. Nilai minimal dari masing-masing pernyataan adalah 1 dan nilai maksimal adalah 6.

Tabel 5.6 Hasil Penilaian Kuesioner Kelas A

No	NRP	Nomor Pernyataan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100051	5	5	5	5	4	6	5	5
2	5113100075	5	5	4	6	5	6	6	6
3	5113100128	5	6	5	5	6	6	5	6
4	5113100172	4	4	3	3	5	5	4	3
5	5115100062	5	6	4	5	5	6	4	6

5.3.2 Pengujian Kegunaan Skenario Kelas B

Berikut ini adalah rincian pengujian kegunaan skenario kelas B:

- Nama Mata Kuliah : Data Mining
- Kelas : A
- Jam pelaksanaan : Pukul 19.30 – 22.30 WIB
- Ruang Kelas : IF-103

Tabel 5.7 adalah tabel hasil penilaian kuesioner skenario kelas B dari masing-masing mahasiswa pada setiap nomor pernyataan kuesioner. Nilai minimal dari masing-masing pernyataan adalah 1 dan nilai maksimal adalah 6.

Tabel 5.7 Hasil Penilaian Kuesioner Kelas B

No	NRP	Nomor Pernyataan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100053	4	4	4	4	4	4	4	4
2	5113100072	5	6	6	6	6	5	6	6
3	5113100143	4	4	4	4	4	4	4	4
4	5113100173	5	5	3	5	3	5	3	5
5	5113100174	5	4	6	2	3	2	3	5

5.3.3 Pengujian Kegunaan Skenario Kelas C

Berikut ini adalah rincian pengujian kegunaan skenario kelas C:

- Nama Mata Kuliah : PBKK (Pemrograman Berbasis Kerangka Kerja)
- Kelas : A
- Jam pelaksanaan : Pukul 22.00 – 23.00 WIB
- Ruang Kelas : Laboratorium Algoritma Pemrograman

Tabel 5.8 adalah tabel hasil penilaian kuesioner skenario kelas B dari masing-masing mahasiswa pada setiap nomor pernyataan kuesioner. Nilai minimal dari masing-masing pernyataan adalah 1 dan nilai maksimal adalah 6.

Tabel 5.8 Hasil Penilaian Kuesioner Kelas C

No	NRP	Nomor Pernyataan Kuesioner							
		1	2	3	4	5	6	7	8
1	5113100127	5	3	4	6	5	5	4	3
2	5113100141	5	6	2	5	4	6	5	6
3	5113100144	5	2	4	3	6	3	3	5
4	5113100146	4	4	6	3	5	5	5	4

No	NRP	Nomor Pernyataan Kuesioner							
		1	2	3	4	5	6	7	8
5	5113100015	4	4	4	4	4	4	4	4

Hasil penilaian dari ketiga kelas dilakukan rekapitulasi untuk mendapatkan nilai rata-rata dari masing-masing pertanyaan kuesioner yaitu dengan menghitung jumlah nilai masing-masing pernyataan kuesioner pada skenario kelas A, B, dan C kemudian dibagi dengan jumlah responden. Hasil rata-rata penilaian masing-masing pernyataan kuesioner dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Penilaian Rata-Rata Kuesioner Seluruh Kelas

No	Pernyataan	Hasil Penilaian (skala 1 – 6)
1	Menurut saya, aplikasi ini mudah untuk digunakan	4.7
2	Menurut saya, presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (presensi di kertas)	4.5
3	Menurut saya, lebih mudah dan efisien melakukan presensi dengan menggunakan aplikasi ini daripada sistem manual (presensi di kertas)	4.3
4	Menurut saya, mudah menggunakan fitur presensi dengan pencocokan wajah	4.4
5	Menurut saya, mudah memasukkan data wajah ke dalam aplikasi	4.6
6	Menurut saya, hasil presensi dengan fitur pencocokan wajah memiliki hasil yang akurat	4.8
7	Menurut saya, presensi dengan menggunakan fitur pencocokan wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (presensi di kertas)	4.3

No	Pernyataan	Hasil Penilaian (skala 1 – 6)
8	Menurut saya, lebih mudah dan efisien melakukan presensi dengan menggunakan fitur pencocokan wajah dalam aplikasi ini daripada sistem manual (presensi di kertas)	4.8

Hasil penilaian rata-rata dari pengujian kegunaan aplikasi yang dilakukan oleh 15 responden menghasilkan rata-rata penilaian mencapai skala 4.7 bahwa aplikasi ini mudah untuk digunakan, skala 4.5 bahwa presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (presensi di kertas), skala 4.3 bahwa lebih mudah dan efisien melakukan presensi dengan menggunakan aplikasi ini daripada sistem manual (presensi di kertas), skala 4.4 bahwa mudah menggunakan fitur presensi dengan pencocokan wajah, skala 4.6 bahwa mudah memasukkan data wajah ke dalam aplikasi, skala 4.8 bahwa hasil presensi dengan fitur pencocokan wajah memiliki hasil yang akurat, skala 4.3 bahwa presensi dengan menggunakan fitur pencocokan wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (presensi di kertas), skala 4.8 bahwa lebih mudah dan efisien melakukan presensi dengan menggunakan fitur pencocokan wajah dalam aplikasi ini daripada sistem manual (presensi di kertas).

5.4 Pengujian Waktu Eksekusi

Pengujian waktu eksekusi ini dilakukan untuk mengetahui tingkat kecepatan aplikasi dalam melakukan proses verifikasi kehadiran pada kasus penggunaan melakukan presensi. Waktu eksekusi dihitung mulai dari pengguna memilih pilihan verifikasi kehadiran hingga dialog konfirmasi kehadiran ditampilkan. Pengguna aplikasi pada pengujian ini sama dengan pengguna pada pengujian kegunaan. Daftar pengguna dapat dilihat pada Tabel 5.5. Hasil pengujian waktu eksekusi dapat dilihat pada Tabel 5.10.

Tabel 5.10 Hasil Pengujian Waktu Eksekusi

No	NRP	Smartphone dan Spesifikasi	Waktu Eksekusi	Kelas
1	5113100051	Sony C5 + Android Marshmallow	6 detik	Kelas A
2	5113100075	Samsung Galaxy Note 3 + Android Marshmallow	5 detik	Kelas A
3	5113100128	Sony Xperia Z5 + Android Nougat	10 detik	Kelas A
4	5113100172	Samsung Galaxy A5 + Android Marshmallow	12.59 detik	Kelas A
5	5115100062	Xiaomi Redmi Note 2 + Android Lollipop	13.7 detik	Kelas A
6	5113100053	Samsung Galaxy J2 + Android Lollipop	8.68 detik	Kelas B
7	5113100072	Oppo A37 + Android Lollipop	12.81 detik	Kelas B
8	5113100143	Samsung Galaxy J2 + Android Lollipop	6.39 detik	Kelas B
9	5113100173	Sony Xperia Z1 Compact + Android Lollipop	14.3 detik	Kelas B
10	5113100174	Oppo A37 + Android Lollipop	33.49 detik	Kelas B
11	5113100127	Samsung Galaxy A3 + Android Marshmallow	7 detik	Kelas C
12	5113100141	Lenovo Vibe K5 ⁺ + Android Lollipop	12.99 detik	Kelas C
13	5113100144	LG Google Nexus 5 + Android Marshmallow	15 detik	Kelas C
14	5113100146	Lenovo Vibe K5 ⁺ + Android Lollipop	17.59 detik	Kelas C

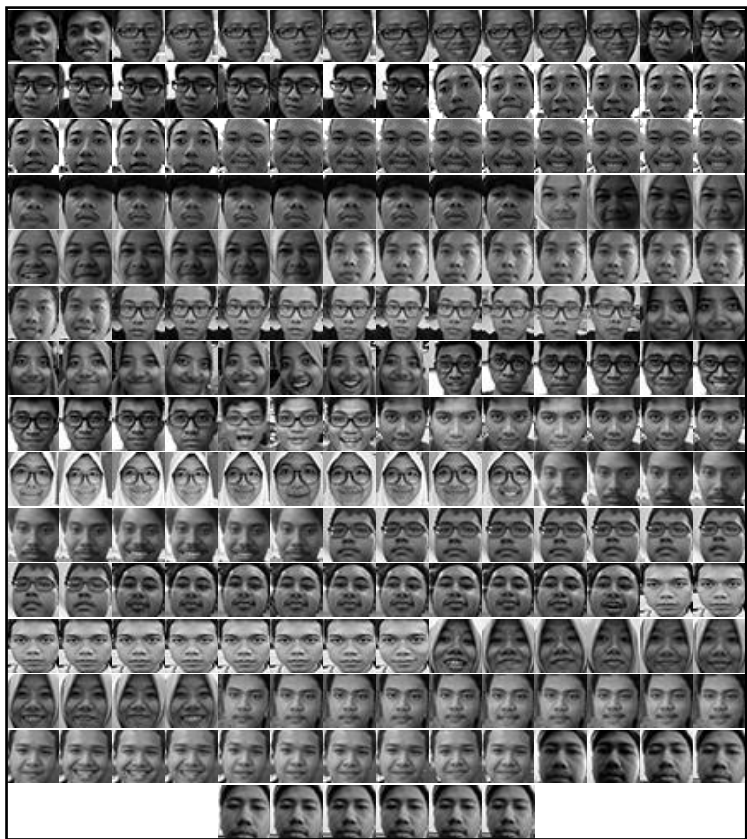
No	NRP	Smartphone dan Spesifikasi	Waktu Eksekusi	Kelas
15	5113100015	Xiaomi 4 + Android Marshmallow	12 detik	Kelas C

Hasil pengujian waktu eksekusi berdasarkan Tabel 5.10 diperoleh perbedaan waktu pada tiap kelasnya. Untuk kelas A, waktu eksekusi tercepat adalah 5 detik dan waktu eksekusi terlama adalah 13.7 detik. Untuk kelas B, waktu eksekusi tercepat adalah 6.39 detik dan waktu eksekusi terlama adalah 33.49 detik. Untuk kelas C, waktu eksekusi tercepat adalah 7 detik dan waktu eksekusi terlama adalah 17.59 detik.

5.5 Pengujian Kapasitas Pengolahan Data Wajah

Pengujian kapasitas pengolahan data wajah merupakan pengujian terhadap kemampuan aplikasi untuk menampung dan mengolah seberapa banyak data wajah yang dapat diolah oleh perangkat *smartphone*. Pada Gambar 5.9 ditunjukkan contoh data wajah dalam 1 kelas. Kelas ini memiliki jumlah mahasiswa sebanyak 30 orang dimana tiap orang memiliki 10 foto wajah yang disimpan dan diolah. Pada Tabel 5.11 ditunjukkan rincian kapasitas yang dibutuhkan oleh tiap data wajah mahasiswa dalam 1 kelas.





Gambar 5.9 Data Wajah Dalam 1 Kelas

Tabel 5.11 Rincian Kapasitas Data Wajah Tersimpan

No	NRP	Nama Peserta	Kapasitas
1	5113100026	Kharisma Nur Annisa	652 kb
2	5113100031	Adian Latifa Nurrohman	719 kb
3	5113100046	Burhanudin Rasyid	577 kb
4	5113100051	Mardiana Sekarsari	679 kb
5	5115100053	Lophita Y. Napitupulu	513 kb

6	5113100072	Ayu Kardina Sukmawati	688 kb
7	5113100075	Achmad Rizky Haqiqi	773 kb
8	5113100077	Ilham Gurat	527 kb
9	5113100090	Didit Sepiyanto	580 kb
10	5113100092	Fajar Ade Putra	515 kb
11	5113100098	Dewangga W Winardi	563 kb
12	5113100112	Ronald Gunawan R	643 kb
13	5113100114	Ken Genesius M B	515 kb
14	5113100121	Cayza Angka Maulana	479 kb
15	5113100127	Luffi Aditya Sandy	580 kb
16	5113100128	Relaci Aprilia Istiqomah	721 kb
17	5113100140	Reza Andriyunanto	634 kb
18	5113100141	Laurensius Adi	691 kb
19	5113100143	Nuning Septiana	665 kb
20	5113100144	Andre Zachary Reinaldi	770 kb
21	5113100146	Kevin Zulkarnain Yuseti	681 kb
22	5113100147	Lusiana Nurul Aini	662 kb
23	5113100151	Danang Adi Nugroho	615 kb
24	5113100172	Afif Ishamsyah Hantriono	814 kb
25	5113100173	Andi Akram Yusuf	761 kb
26	5113100174	Armirara Refa Aziz	713 kb
27	5114100019	Sita Nurjayanti	539 kb
28	5114100153	Andreas Galang Anugerah	619 kb
29	5114100163	Muhammad Habibur Rahman	576 kb
30	5115100062	Alvin Mudhoffar	678 kb
Total			13827 kb

Berdasarkan rincian kapasitas data wajah tersimpan pada Tabel 5.11, aplikasi membutuhkan kapasitas memori kosong sebesar 13.927 kb (*kilobytes*) atau setara 13,5 mb (*megabytes*) untuk 1 kelas dengan total peserta sebanyak 30 anak.

5.6 Evaluasi Pengujian Fungsionalitas

Berdasarkan pengujian yang telah dilakukan, terdapat pengujian fungsionalitas memberikan hasil yang sesuai dengan skenario yang direncanakan dan juga memberikan hasil yang belum sesuai dengan skenario yang direncanakan. Evaluasi

terhadap pengujian yang telah berhasil dilaksanakan sebagai berikut:

1. Fungsionalitas mendaftarkan data wajah berjalan sesuai dengan yang diharapkan. Pengujian UJ.UC-001 memperlihatkan bahwa sistem memberikan respon yang sesuai dengan skenario pengujian.
2. Fungsionalitas melakukan presensi berjalan sesuai dengan yang diharapkan. Pengujian UJ.UC-002 memperlihatkan bahwa sistem memberikan respon yang sesuai dengan skenario pengujian.

5.7 Evaluasi Pengujian Kegunaan

Berdasarkan hasil kuesioner pada Tabel 5.9 ditunjukkan bahwa rata-rata penilaian responden terhadap kegunaan dari aplikasi ini menghasilkan nilai diatas skala 4 sehingga dapat diambil kesimpulan bahwa aplikasi cukup mudah digunakan dan dapat memberikan manfaat bagi pengguna. Manfaat yang diharapkan dari aplikasi ini adalah aplikasi dapat menggantikan sistem kehadiran mahasiswa yang masih manual yang menggunakan presensi di kertas dengan aplikasi ini.

5.8 Evaluasi Pengujian Waktu Eksekusi

Berdasarkan hasil pengujian waktu eksekusi pada Tabel 5.10, terdapat perbedaan waktu dari masing-masing *smartphone* yang digunakan oleh tiap pengguna. Waktu eksekusi tercepat dari semua kelas yang diperoleh adalah selama 5 detik dan waktu terlama menunjukkan 33.49 detik. Hal ini dapat terjadi karena perbedaan dari spesifikasi *smartphone* yang digunakan disertai dengan kondisi pencahayaan ruangan yang tidak optimal. Dari pengujian ini, *smartphone* yang menghasilkan waktu eksekusi tercepat adalah Samsung Galaxy Note 3 dengan sistem operasi Android Marshmallow sedangkan *smartphone* yang menghasilkan waktu eksekusi terlambat adalah Oppo A37 dengan sistem operasi Android Lolipop. Waktu eksekusi tercepat didapatkan dengan spesifikasi *smartphone* berkapasitas memori 32 GB, RAM 3 GB,

kamera depan 2 MP, dan kecepatan prosesor mencapai 1,9 GHz sedangkan waktu eksekusi terlambat didapatkan dengan spesifikasi *smartphone* berkapasitas memori 16GB, RAM 2 GB, kamera depan 5 MP, dan kecepatan proses mencapai 1,2 GHz.

Dari pengujian ini diperoleh batasan bahwa aplikasi memperoleh hasil paling optimal ketika berjalan pada *smartphone* yang memiliki spesifikasi lebih tinggi dari spesifikasi minimal *smartphone* yang sudah ditentukan serta kondisi pencahayaan ruangan untuk kasus penggunaan melakukan presensi yang telah ditentukan dalam batasan masalah dari aplikasi ini.

5.9 Evaluasi Pengujian Kapasitas Pengolahan Data Wajah

Berdasarkan hasil rincian kapasitas data wajah tersimpan pada Tabel 5.11 ditunjukkan bahwa minimal memiliki kapasitas memori kosong sebanyak 13,5 mb untuk menampung data wajah dari seluruh mahasiswa/pengguna dalam 1 kelas. Pengujian ini mengasumsikan 1 pengguna memiliki 10 foto wajah yang diolah sehingga jika terdapat 30 pengguna maka terdapat 300 foto wajah yang akan diolah menjadi data *training* untuk proses verifikasi wajah oleh aplikasi. Dalam pengujian tidak terjadi kesalahan dalam pengolahan foto wajah serta seluruh data wajah berhasil diolah oleh aplikasi. Dari pengujian ini dapat disimpulkan bahwa tidak ada batasan untuk jumlah foto wajah yang dapat diolah oleh aplikasi. Batasan dari pengujian ini terletak dari ketersediaan kapasitas memori yang dimiliki oleh *smartphone*.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditunjukkan untuk pengembangan perangkat lunak di masa mendatang.

6.1 Kesimpulan

Kesimpulan dari tugas akhir ini adalah:

1. Sistem kehadiran mahasiswa melalui pencocokan foto wajah berhasil diimplementasikan sehingga dapat menggantikan sistem verifikasi kehadiran yang secara manual.
2. Pustaka (*library*) Android Face Recognition with Deep Learning pada pencocokan foto wajah berhasil diimplementasikan dan diuji dengan kode pengujian UJ.UC-001 dan UJ.UC-002.

6.2 Saran

1. Perlu dibuat mekanisme khusus supaya data wajah mahasiswa yang terambil lebih variatif / ekspresi wajah tidak monoton sehingga memperkecil kemungkinan terjadinya redundansi data.
2. Perlu diimplementasikan algoritma yang lebih baik yang dapat mengatasi masalah akurasi pengenalan wajah di ruang yang kurang mendapatkan pencahayaan yang optimal.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] “Teknik Informatika ITS.” [Daring]. Tersedia pada: <http://if.its.ac.id/>. [Diakses: 30-Mei-2017].
- [2] “Sistem,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 20-Des-2016.
- [3] “Arti kata hadir - Kamus Besar Bahasa Indonesia (KBBI) Online.” [Daring]. Tersedia pada: <http://kbbi.web.id/hadir>. [Diakses: 29-Des-2016].
- [4] Metode-Algoritma.com, “Face Detection dan Face Recognition Eigenface | Contoh Program.”.
- [5] “Qualeams/Android-Face-Recognition-with-Deep-Learning-Library,” *GitHub*. [Daring]. Tersedia pada: <https://github.com/Qualeams/Android-Face-Recognition-with-Deep-Learning-Library>. [Diakses: 20-Apr-2017].
- [6] “OpenCV | OpenCV.” [Daring]. Tersedia pada: <http://opencv.org/>. [Diakses: 11-Des-2016].
- [7] “OpenCV: Face Detection using Haar Cascades.” [Daring]. Tersedia pada: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html. [Diakses: 19-Jun-2017].
- [8] “Feature extraction,” *Wikipedia*. 12-Mei-2017.
- [9] “LIBSVM -- A Library for Support Vector Machines.” [Daring]. Tersedia pada: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>. [Diakses: 20-Apr-2017].
- [10] “Application programming interface,” *Wikipedia*. 06-Des-2016.
- [11] “JSON,” *Wikipedia*. 11-Des-2016.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

SIKEMAS

- Sistem Informasi Kehadiran Mahasiswa -

Aplikasi dibuat untuk memenuhi Tugas Akhir S1 Teknik Informatika ITS. Aplikasi ini bertujuan untuk memberikan solusi terhadap kecurangan (Titip Absen) dalam melakukan presensi kehadiran mahasiswa di perkuliahan. Ruang lingkup aplikasi ini adalah Teknik Informatika ITS.

*** Diperlukan**

1. Menurut saya, aplikasi ini mudah untuk digunakan *

Tandai hanya satu oval.

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat Setuju

2. Menurut saya, presensi kehadiran dengan menggunakan aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas)

Tandai hanya satu oval.

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat Setuju

3. Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan aplikasi ini daripada sistem manual (absen di kertas) *

Tandai hanya satu oval.

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat Setuju

4. Menurut saya, tidak sulit menggunakan fitur presensi kehadiran dengan foto wajah *

Tandai hanya satu oval.

1 2 3 4 5 6

Tidak Setuju

Sangat Setuju

5. Menurut saya, tidak sulit memasukkan dataset foto wajah ke dalam aplikasi *

Tandai hanya satu oval.

[illegible]







6. Menurut saya, hasil presensi kehadiran dengan fitur foto wajah memiliki hasil yang akurat *

Tandai hanya satu oval.

[illegible]

7. Menurut saya, lebih mudah dan efisien melakukan presensi kehadiran dengan menggunakan fitur foto wajah dalam aplikasi ini daripada sistem manual (absen di kertas) *

Tandai hanya satu oval.

	1	2	3	4	5	6	
Tidak Setuju							Sangat Setuju

8. Menurut saya, presensi kehadiran dengan menggunakan fitur foto wajah dalam aplikasi ini lebih aman dari kecurangan daripada sistem manual (absen di kertas) *

Tandai hanya satu oval.

	1	2	3	4	5	6	
Tidak Setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sangat Setuju

9. Kritik dan saran mengenai aplikasi presensi kehadiran ini *

10. Kritik dan saran mengenai fitur presensi kehadiran menggunakan fitur foto wajah *

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Ronald Gunawan R atau yang biasa dipanggil dengan Ronald di lingkungan kampus, lahir pada tanggal 10 November 1994 di Semarang. Penulis menempuh pendidikan SMA di SMA Kolese Loyola Semarang dan S1 Teknik Informatika Fakultas Teknologi Informasi di Institut Teknologi Sepuluh Nopember Surabaya. Sejak masuk kuliah informatika, Penulis mulai belajar dengan yang namanya pemrograman, meskipun sulit karena tidak ada pengalaman di pemrograman sebelumnya. Sejak tahun ke-2 perkuliahan, Penulis mulai tertarik dengan pemrograman berbasis objek dan mulai mempelajari pemrograman perangkat bergerak. Penulis memperoleh pengalaman dalam hal pemrograman dari tugas-tugas kuliah dan beberapa proyek dari luar yang dilakukan bersama dengan beberapa teman kuliah. Penulis dapat dihubungi melalui alamat surel **geronald10@gmail.com**.

